Volume : 3 | Issue : 7 | Received-27-06-2026 | Accepted-02-07-2026 | Published-07-07-2026

UAI JOURNAL OF ARTS, HUMANITIES AND SOCIAL SCIENCES (UAIJAHSS)

Abbreviated Key Title: UAI J Arts Humanit Soc Sci
ISSN: 3048-7692 (Online)
Journal Homepage: https://uaipublisher.com/uaijahss/
Volume- 3 Issue- 7 (July) 2026
Frequency: Monthly

A Preliminary Exploration of the Knowledge Base for Excavated Documents Research

Tung Kiu Kit[1], Ho Yin Gary YEE[2], Wen-Chuan Ke[3], Jheng-Yu Yin[4]

[1] Ph D., East China Normal University; Lecturer, Hong Kong Metropolitan University.

[2] Ph D., East China Normal University; Chairman, Silk Road Cultural Society.

[3] Ph D. Researcher Fellow, School of Technology Management, National Yang Ming Chiao Tung University; Adjunct Associate Professor, Department of Industrial Engineering and Management, National Taipei University of Technology; Adjunct Associate Professor, Department of Public Administration, National Open University; Visiting Professor, School of Intelligent Construction, Wuchang University of Technology.

[4] Ph D., Fu Jen Catholic University; Visiting Professor, Normal College, Hubei Polytechnic University.

Corresponding Author: Tung Kiu Kit

ABSTRACT

This article focuses on the application and transformation of artificial intelligence technology in the construction of knowledge bases for excavated document research. It systematically traces the evolution of excavated document knowledge bases from early digitized databases, knowledge graphs, and multimodal databases to new paradigms such as Retrieval Augmentation Generation (RAG), GraphRAG, Agentic RAG, and LLM-Wiki. This paper argues that while traditional RAG can improve the efficiency of document retrieval and question answering, it still faces limitations in research tasks unique to excavated documents, such as fragmented retrieval, insufficient context restoration, weak knowledge connections, and difficulties in version maintenance, particularly in areas like intertextual relationships, diachronic evolution, character shape comparison, historical interpretation, and cross-material connections. In contrast, the LLM-Wiki methodology proposed by Andrej Karpathy in 2026 emphasizes replacing temporary splicing during queries with incremental compilation, persistent maintenance, and structured connections, providing a new approach for knowledge bases to shift from "instant question answering tools" to "sustainably growing research-oriented knowledge systems." This paper analyzes the advantages and disadvantages of different tools, including IMA, WeKnora, MaxKB, and nashsu / llm_wiki, in terms of application threshold, knowledge organization, academic adaptability, and human-computer collaboration. The study argues that future construction of excavated document knowledge bases should adopt a fusion approach of "RAG instant retrieval + LLM-Wiki long-term accumulation," and further focus on multimodal understanding, structured memory, knowledge entropy control, and prevention of recursive distortion, in order to construct a digital humanities research infrastructure that combines traceability, evolvability, and academic rigor.

In contrast, the LLM-Wiki methodology, proposed by Andrej Karpathy in 2026, emphasizes incremental compilation, persistent maintenance, and structured linkage over ad-hoc query-time stitching. This approac h offers a new paradigm for

DOI: 10.5281/zenodo.21236315

transitioning knowledge bases from "real-time Q&A tools" to "sustainable, evolving research knowledge systems." Furthermore, this paper analyzes several existing frameworks—including IMA, WeKnora, MaxKB, and nashsu / llm_wiki — evaluating their relative strengths and weaknesses regarding accessibility, knowledge organization, academic adaptability, and human-AI collaboration.

The study proposes that future development of excavated documents knowledge bases should adopt a hybrid pathway of "RAG for real-time retrieval + LLM-Wiki for long-term consolidation." It further emphasizes the need to prioritize advancements in multimodal understanding, structured memory, knowledge entropy control, and the mitigation of recursive distortion. Ultimately, these efforts aim to establish a digital humanities research infrastructure characterized by traceability, evolvability, and academic rigor.

KEY WORDS: Artificial Intelligence; Excavated Documents; Knowledge Base; LLM-Wiki; Retrieval-Augmented Generation.

1. Introduction

In August 2025 , the State Council issued the "Opinions on Deepening the Implementation of the ' Artificial Intelligence + ' Action," clearly proposing the strategic goal of "fully entering the intelligent economy and intelligent society by 2035," and taking "strengthening the interdisciplinary driving role of artificial intelligence " as the core orientation for the restructuring of university disciplines. Against this backdrop, the study of excavated documents, as a fundamental discipline for the inheritance of excellent traditional Chinese culture, has ushered in unprecedented opportunities and challenges empowered by technology. Breakthroughs in artificial intelligence technology, especially deep learning, natural language processing, and large language models (LLM), have provided entirely new tools for the interpretation, organization, and pattern discovery of excavated documents. The emergence of large language models has transformed personal knowledge bases from static data stacks into dynamic knowledge systems, capable of automatically sorting out knowledge threads, connecting related content, and even generating new research clues based on existing knowledge, breaking through the boundaries of traditional knowledge organization. However, the construction of a dedicated LLM-driven knowledge base for the field of excavated documents still requires further exploration by the academic community to find more suitable solutions.

This paper aims to: (1) review the current application status and theoretical limitations of artificial intelligence knowledge base technology in the study of excavated documents; (2) analyze the multidimensional dilemmas in the construction of excavated document knowledge bases in combination with the latest technologies; and (3) introduce the LLM-wiki methodology proposed by Andrej Karpathy to demonstrate its adaptability and innovative value in the field of excavated documents.

Breakthroughs in artificial intelligence technologies — particularly in deep learning, natural language processing (NLP), and Large Language Models (LLMs) — have provided novel tools for the interpretation, collation, and pattern discovery within excavated documents. The emergence of LLMs has transformed personal knowledge bases from static repositories of data into dynamic knowledge systems. These systems can automatically organize contextual knowledge frameworks, associate related content, and even evolve to generate new research leads based on existing knowledge, thereby breaking through the boundaries of traditional knowledge organization. However, in this context, the construction of exclusive, LLM-driven knowledge bases specifically tailored to the field of excavated documents remains an area requiring further exploration and the development of more suitable adaptive solutions by the academic community.

This paper aims to: (1) review the current applications and theoretical limitations of AI knowledge base technologies in excavated documents research; (2) analyze the multidimensional challenges in constructing excavated documents knowledge bases in light of contemporary technological advancements; and (3) introduce the LLM-Wiki methodology proposed by Andrej Karpathy, arguing for its adaptability and innovative value within the field of excavated documents research.

2. Evolution and Development of Intelligent Knowledge Base Technology

2.1 From Digitalization to Intelligentization: The Technological Evolution of Knowledge Bases

The digitization of documentary research and the construction of knowledge bases can be traced back to the development of documentary databases in the 1990s. Early landmark achievements included large-scale full-text digitization of ancient books, such as the electronic version of the *Siku Quanshu* (Complete Library of the Four Treasuries). In 2002, the *Basic Ancient Books Database of China * [1]project was conceived, compiled, and supervised by Liu Junwen of Peking University. Developed and produced by Beijing Airusheng Company, it includes approximately 10,000 ancient books, with about 1.7 billion characters in full text and about 10 million pages of images , roughly equivalent to three *Siku Quanshu*, making it the largest full-text digitization project of ancient books at the time.[2]

Since the beginning of the 21st century, with the maturation of image recognition and natural language processing technologies, knowledge bases have begun to evolve towards "intelligent association." Currently, excavated document knowledge bases mainly exhibit three technological paths:

[1] Airosheng official website: http://er07.com/home/pro_3.html

[2] Geng Yuanli, “A Review of Research on the Digitization of Ancient Chinese Books in the Past Thirty Years (1979-2009)”, Guoxue.com, August 18, 2009, http://www.guoxue.com/wk/000652.htm?u_atoken=69fa036d-f409-3204-2ad0-36b4944c4b3a&u_asig=76fdacb517779925574694711e

DOI: 10.5281/zenodo.21236315

First, a knowledge graph based on semantic annotation, such as the "Comprehensive Digital Platform for Chinese Bamboo and Silk Manuscripts," which attempts to structurally link elements such as bamboo slip texts, dictionaries, textual research, and documents.

Second, there are multimodal databases that integrate heterogeneous data from multiple sources. For example, the "Yinqi Wenyuan" platform integrates rubbings, copies, transcriptions, and research literature of oracle bone inscriptions, and provides AI tools such as image recognition and automatic joining.

Third, there are dynamic knowledge networks built on deep learning models, such as the open-source project "DMOB (Dynamic Mosaic Oracle Bone Dataset)," which generates diverse oracle bone script images through algorithms to support the training of object detection models.

At the retrieval technology level, Retrieval-Augmented Generation (RAG) has become the mainstream paradigm for knowledge base retrieval, with most intelligent knowledge bases currently using RAG technology as their underlying foundation. RAG improves question-answering accuracy in specific scenarios by combining large language models with domain knowledge bases, without requiring model retraining. Its core process includes three stages: document processing, data retrieval, and augmentation generation.

2.2 Retrieval Enhancement Generation (RAG) Technology Lineage and LLM- Wiki Paradigm: Timeline Phases and Technology Comparison

Retrieval-Augmented Generation (RAG) can be traced back to the early 1970s , when researchers began exploring information retrieval and natural language processing (NLP) in hopes of developing systems capable of answering questions in natural language. These early systems, while limited in scope, laid the foundation for subsequent text mining and information retrieval. The introduction of the natural language [3]question-answering system by Ask Jeeves in the mid-1990s, culminating in IBM Watson's victory in the Jeopardy! Question-Answering competition in 2011, marked a crucial turning point for RAG technology towards practical application. This technology has evolved from "general retrieval" to "agent collaboration."

This section will use time as a timeline to trace the development of knowledge base technology from a historical perspective, and introduce the LLM-wiki methodology proposed by Andrej Karpathy to explore its differences and technological value in the construction of knowledge bases for excavated documents.

2.2.1First Phase: Foundation Laying Period (2020-2022)

This stage is characterized by the original retrieval enhancement generative framework proposed by Lewis et al. in 2020, 4whose core processing flow is "document slicing → vectorization → vector retrieval → prompt concatenation → answer generation". The design philosophy of this framework aims to compensate for the limitations of large-scale language models caused by the inability to update pre-trained data in a timely manner and the lack of domain knowledge by introducing external knowledge base materials. Key technical components cover document chunking, vectorization encoding, approximate nearest neighbor retrieval, and prompt engineering. Document chunking typically employs fixed-length or sliding window strategies to segment long texts into semantically coherent units; vectorization encoding utilizes pre-trained models such as BERT and RoBERTa to map text to a high-dimensional vector space; approximate nearest neighbor retrieval relies on libraries such as FAISS and Annoy to achieve high-level and fast vector similarity search; and prompt engineering concatenates the retrieval results with the user's question to form contextual input for the generative model.

2.2.2 Second Phase: Advanced Optimization Period (2023-2024)

With the widespread application of large-scale models, researchers have proposed several improvements to address the shortcomings of naive RAG, forming the "Advanced RAG" paradigm. The core of this stage lies in the system optimization of the pre-retrieval, during-retrieval, and post-retrieval stages. Pre-retrieval optimization includes semantic chunking, query rewriting, and multi-granularity indexing. Semantic chunking dynamically segments documents based on sentence boundaries, paragraph structure, or thematic coherence to preserve complete semantics; multi-granularity indexing simultaneously constructs character-level, word-level, sentence-level , and document-level indexes to support precise recall at different levels of detail . During-retrieval optimization encompasses hybrid retrieval, re-ranking, and multi-hop retrieval. Hybrid retrieval combines sparse retrieval (BM25 keyword matching) and dense retrieval (vector similarity), balancing precise matching and semantic generalization; re-ranking uses models such as Cross-Encoder to perform secondary ranking of the initial screening results , improving the relevance of Top-K results; multi-hop retrieval mines implicit associations through repeated query operations. Post-retrieval optimization includes citation tracing, answer calibration, and feedback learning. Citing sources and forcing the generation of answers to include evidence sources allows for one-click verification of original literature; answer calibration introduces confidence scoring and multi-model voting mechanisms to reduce the risk of single-model hallucination; feedback learning collects scholars' feedback on the generated results to optimize retrieval strategies. Representative technologies such as Self-RAG introduce introspective markers to dynamically determine retrieval timing and answer critique, suitable for verifying the evidence chain of disputed characters, but with high latency; Corrective RAG (CRAG) automatically switches to external network retrieval when the internal knowledge base is insufficient, which can supplement the real-time information of newly excavated bamboo and silk manuscripts , but needs to be protected from network noise; Adaptive RAG dynamically selects retrieval strategies based on query complexity, adapting to different tasks such as "character shape query" and "ideological origin tracing" in excavated document research. This stage of technology has significantly improved the retrieval accuracy and answer reliability of excavated

[3] AskJeeves (later renamed Ask.com) is a commercial search engine and question-and-answer service founded in 1996 by Garrett Gruener and David Warthen , and officially launched in 1997. Its core business concept is to allow users to search using sentences (natural language) rather than keywords. For more details, see: https://tavadiscovery.com/2023/09/ask-jeeves-search-engine-the-first-with-plain-language-queries-and-natural-language-processing/

[4]Augmented Generation) framework, proposed and named by Lewis et al. in 2020 , is generally regarded as a landmark work marking the formal formation of RAG "retrieval-augmented generation" . Lewis , *P. , Perez , E. , Piktus , A. , Petroni , F. , Karpukhin , V. , Goyal , N. , Küttler , H. , Lewis , M. , Yih, W.-t. , Rocktäschel , T. , Riedel , S. , & Kiela , D. (2020) . Retrieval-augmented generation for knowledge-intensive NLP tasks.arXiv.* https://doi.org/10.48550/arXiv.2005.11401

DOI: 10.5281/zenodo.21236315

document knowledge bases, but still faces the fundamental challenges of "information silos" and "weak connections," prompting the academic community to explore new paths empowered by graph structures.

Platforms began to integrate RAG capabilities. For example, Dify, an open-source platform designed specifically for large language model (LLM) applications, made a major upgrade to RAG in version v0.3.31. It implemented key RAG functions through hybrid retrieval, re-ranking models, and multi-path retrieval, and achieved a 20% improvement in retrieval hit rate and an 18.44% increase in the Ragas evaluation framework. In some scenarios, it even surpassed the performance improvement results of OpenAI 's Assistants API at that time, [5]marking the maturity of RAG capabilities.

2.2.3 Phase Three: Retrieval Enhancement and Modularization Period (2024-2025)

Since 2024, the maturity of graph neural networks and knowledge graph technologies has driven the leap from "retrieval-enhanced generation" (RAG) to "structured semantic understanding." The technology has iterated rapidly, resulting in representative frameworks such as RAGFlow, GraphRAG, LightRAG, and Graphiti. Meanwhile, the modular design concept allows RAG systems to combine retrieval, reasoning, and generation components as needed, greatly improving engineering flexibility.

RAGFlow is an open-source search enhancement and generation engine. Its core purpose is to improve the accuracy and reliability of AI question answering and reduce "hallucinations" by integrating external knowledge bases to provide accurate context for large language models. Its Deep Document Understanding [6]feature is a major reason why many users choose it. It abandons off-the-shelf intermediary software and has developed its own processing system based on machine vision and text parsing technology. It can accurately parse the complex layout structure of PDFs, docxes, ppts, and even images, ensuring the quality of data loaded into the knowledge base.

GraphRAG, proposed by Microsoft in 2024,[7]aims to address the shortcomings of traditional vector retrieval in global understanding, relational reasoning, and structured summarization. It is particularly suitable for scenarios requiring in-depth analysis of complex document sets, discovery of hidden information, or answering comprehensive questions. Its core lies in the deep integration of large language models and knowledge graphs, enhancing the contextual understanding capabilities of retrieval and generation through structured semantic networks. This technology first uses a large language model to automatically extract entities and relations from unstructured text, constructing a global knowledge graph. Then, it uses community detection algorithms such as Leiden to form semantic clusters, allowing the retrieval process to focus on highly relevant "knowledge communities." Its advantage lies in its ability to effectively handle complex, abstract, or multi-hop reasoning problems; however, graph construction relies on batch processing and lacks real-time incremental updates, making it suitable for static or low-frequency updated knowledge base scenarios.

LightRAG[8]emphasizes lightweight design and high efficiency, specifically designed for resource- constrained or rapidly deployable scenarios. It employs a two-layer retrieval mechanism: the bottom layer retrieves specific fragments based on vector similarity, while the upper layer utilizes a graph structure to capture broad-domain topic associations. LightRAG achieves superior question-answering accuracy compared to traditional RAGs while maintaining low computational resource consumption. However, its knowledge graph is constructed in a one-time manner and does not support automatic perception and dynamic fusion of new data, making it difficult to address the domain requirements of high-frequency knowledge evolution. In applications involving excavated documents, LightRAG can complete cross-batch association retrieval of tens of thousands of bamboo slips in milliseconds, making it suitable for real-time interaction on museum mobile devices or in resource-constrained scenarios.

Graphiti, another important framework in this phase, focuses on the incremental updating capability of dynamic graphs, greatly compensating for the shortcomings of GraphRAG and LightRAG in real-time knowledge fusion . Its streaming architecture automatically triggers entity extraction and relation completion when new excavated documents are imported, enabling progressive expansion of the graph. For example, when a new batch of Warring States Chu bamboo slips is released, Graphiti can identify personal names, place names, and object names within hours and establish associations with relevant entries in the existing knowledge base, forming a dynamically updated semantic network. The implementation of modular design in this phase provides a highly customized toolchain for excavated document research. Scholars can flexibly combine different functional modules according to specific task requirements. For example, for oracle bone script character comparison tasks, an "image feature extraction module + vector retrieval module + character evolution graph module" can be integrated; for tasks involving tracing the ideological origins of bamboo and silk manuscripts, a " multi-step reasoning module + knowledge graph retrieval module + textual interpretation association module" can be used. This modular approach not only lowers the technical barrier but also allows scholars of excavated documents who are not computer professionals to quickly build intelligent tools that meet their own research needs.

At the practical application level, this stage of technology effectively solves the problem of "cross-source association" in the study of excavated documents. For example, using GraphRAG's wide-area detection algorithm, related legal provisions scattered in the

[5]JasonLee1211: " Dify : How to Efficiently Build Production-Grade Agentic AI Applications?", ITbang Help Network article , October 1, 2025, https://ithelp.ithome.com.tw/articles/10391865

[6]RAGFlow official documentation and usage guide: https://ragflow.io/docs/

[7]Edge , D. , Trinh , H. , Cheng , N. , Bradley , J. , Chao , A. , Mody , A. , Truitt , S. , & Larson , J.(2024).Fromlocaltoglobal:AgraphRAGapproachtoquery-focusedsummarization.arXiv.https://doi.org/10.48550/arXiv.2404.16130

[8]LightRAG (SimpleandFastRetrieval-AugmentedGeneration) is a retrieval augmentation generation (RAG) system published in October 2024 by a research team from the University of Hong Kong and Beijing University of Posts and Telecommunications. This system primarily addresses the pain point of traditional RAG systems struggling to handle complex queries across multiple timeframes or macroscopic scales, and improves upon the high computational cost of Microsoft GraphRAG . For details, see: Guo , Z. , Xia , L. , Yu , Y. , Ao , T. , & Huang , C. (2024). LightRAG:Simpleandfastretrieval-augmentedgeneration.arXiv. https://doi.org/10.48550/arXiv.2410.05779

DOI: 10.5281/zenodo.21236315

Shuihudi Qin bamboo slips, Zhangjiashan Han bamboo slips, and Yinqueshan Han bamboo slips can be semantically clustered to form a "Qin and Han legal system" knowledge community, helping scholars to intuitively study the institutional inheritance relationship between different bamboo slips ; while LightRAG's two-layer retrieval mechanism supports quick querying of the frequency of occurrence and evolution trajectory of a specific character form in different excavated documents on mobile terminals, providing convenience for real-time verification in field investigations.

This stage of technological evolution marks the transformation of the excavated document knowledge base from a "static retrieval tool" to a "dynamic knowledge collaboration platform," laying a solid foundation for the subsequent development of intelligent agent collaboration and context engineering.

2.2.4Phase Four: Agent Collaboration and Context Engineering Phase (2025-2026)

Of multi-agent systems and context engineering[9], as of April 2026, RAG technology has shown two major divergent paths: one is "Agentic RAG," evolving towards proactive reasoning[10]; the other is the LLM-wiki methodology, transforming towards "knowledge compilation." The core feature of Agentic RAG is embedding retrieval into multiple agent loops. The system can autonomously plan retrieval strategies, evaluate answer quality, and iteratively optimize queries, achieving a closed-loop enhancement of "retrieval — reasoning — verification. "Key technologies include proactive retrieval, iterative refinement, multi-source heterogeneous fusion, and tool invocation. Active retrieval allows the agent to autonomously determine the timing, source of information, and depth of retrieval based on the complexity of the problem. For example, it can initiate a three-way search using a character shape database, a dictionary database, and academic history for difficult characters. Iterative refinement automatically assesses the sufficiency of evidence after generating a preliminary answer, triggering a secondary search or expert consultation if insufficient. Multi-source heterogeneous fusion simultaneously searches internal knowledge bases, academic databases, and online resources, and performs credibility-weighted integration. Tool calls integrate specialized tools such as OCR recognition, image comparison, and phonological query to expand the search dimensions. Regarding its application prospects in excavated documents, the intelligent affixing assistant can analyze the edge features of fragments, semantic examples, and calligraphic styles to generate candidate affixing schemes and evaluate their matching degree. The interpretation dispute arbitration can automatically collect different scholars' interpretations of a character, compare the strength of evidence, and generate a neutral review for researchers' reference. Cross-material association can link oracle bone inscriptions, bronze inscriptions, and bamboo and silk manuscript databases to explore the evolutionary patterns of the same character shape in different carriers. However, Agentic RAG faces challenges such as high computational overhead, a black box in the inference process, and unclear academic responsibility. It is necessary to design a transparent mechanism and a human-machine rights and responsibilities framework.

2.2.5 Phase 5: Persistent Wiki Maintenance (2025-2026)

The inefficient "temporary book flipping" model of traditional RAG (Retrieval-Augmented Generation), Andrej Karpathy proposed the LLM-wiki (Large Model Autonomous Compilation and Maintenance Persistent Knowledge Base) methodology in April 2026. Its core idea is to allow the large model to act as a "full-time librarian," continuously compiling raw data into a structured, linkable personal knowledge base. Through four steps—raw data input → large language model compilation → wiki → question answering—a unique three-layer architecture is constructed . Taking excavated documents as an example: the raw data layer stores immutable images of bamboo slips, transcriptions, and research papers; the compilation layer ingests data through a two-step thought chain , first having the LLM read the data and output "key entities," "concepts," "relationships with existing knowledge," and "structured analysis of contradictions," then generating wiki pages based on the analysis results, updating the index, and marking projects requiring manual review; the knowledge layer is presented in a structured Markdown web format, supporting bidirectional links and graph browsing. Its four major innovative mechanisms include incremental compilation, four-signal association, asynchronous review, and persistent maintenance. Incremental compilation only processes documents with changed content (SHA256 verification), avoiding redundant calculations and significantly saving computing power ; four-signal association calculates page relevance through four dimensions: direct links, source overlap, Adamic -Adar algorithm, and type affinity , constructing a dynamic knowledge graph; asynchronous review sets up a queue for pending review, allowing scholars to address questionable projects marked by the system in a timely manner; persistent maintenance ensures that knowledge is compiled only once and continuously updated, forming a sustainable human-machine collaborative model. It is worth noting that after the original data is ready, the large language model will incrementally compile a wiki containing different .md files. The large language model will write summaries for the original data, classify the data into concepts, write an article for each concept, establish backlinks between articles, and automatically maintain the page relationship network. As data accumulates, the knowledge system will gradually "grow" more complete and its connectivity will improve.[11]

Compared to traditional RAGs, LLM-wikis differ fundamentally in knowledge acquisition, update mechanisms, human-machine roles, evidence tracing, and applicable scenarios: Traditional RAGs retrieve information on an ad-hoc basis during queries, while LLM-wikis are pre-compiled and persistently stored; traditional RAGs require index rebuilding or incremental vector updates, while LLM-wikis employ incremental compilation and dynamic graph fusion; traditional RAGs involve human questioning and machine answering, while LLM-wikis involve human curation, machine maintenance, and initial screening; traditional RAGs include citations upon generation, while LLM-wikis provide page-level bidirectional links and version history; traditional RAGs are suitable

[9]"Context engineering" can be understood as not only designing prompts, but also systematically designing all the information the model can obtain before and during a certain step of reasoning, including instructions, external knowledge, memory, tools, state, and dialogue context. The goal is to ensure the model obtains the right information at the right time, thereby improving task performance and reliability. For details, please refer to: Hua , Q. , Ye , L. , Fu , D. , Xiao , Y. , Cai , X. , Wu , Y. , Lin , J. , Wang , J. , & Liu , P. (2025). Contextengineering 2.0: The context of contextengineering.arXiv. https://doi.org/10.48550/arXiv.2510.26493

[10]

[11]Elponcrab : " Karpathy Reveals: The Complete Method for Building a Personal Knowledge Base with LLM " , ChainNews, May 4, 2025, https://abmedia.io/karpathy-llm-knowledge-base-personal-wiki-obsidian-method-2026

DOI: 10.5281/zenodo.21236315

for immediate question answering and rapid exploration, while LLM-wikis are better suited for in-depth research, knowledge accumulation, and academic writing. In short, RAGs require re-finding the answer with each query; LLM-wikis are compiled once and maintained persistently.

In terms of adaptability to excavated document research, LLM-wiki excels in automating the sorting and compilation of "interpretive history." An excavated document or fragment of bamboo slips often undergoes interpretation and verification by multiple generations of scholars, resulting in numerous differing interpretations and authentication attempts. Traditional databases mostly only list these interpretations in tables or static text, making it difficult to trace their logical evolution. Researchers can place their published interpretation papers into the raw/(original data) folder of LLM Wiki. LLM acts as a "compiler," automatically reading these papers, generating independent Markdown pages for each "character" or "shortcut," automatically summarizing the "interpretive history summary" for that character, listing points of contention, and even identifying contradictory statements by scholars, automatically marking them on the page (linting). This significantly reduces the "entropy increase" and labor costs for scholars conducting document reviews. Furthermore, there are intricate intertextual relationships between excavated documents and transmitted classics, as well as between different excavated materials. In the past, establishing these connections relied heavily on manual annotation of "knowledge graphs" or entity associations. Under the LLM Wiki framework, LLM automatically detects "concepts" in different document fragments, such as specific official titles, institutional systems, and variant characters, and automatically establishes backlinks. Imagine that when you are reading the page of "Shuihudi Qin Bamboo Slips: Fengzhenshi ", the system has already automatically linked to records of the same legal system in transmitted documents, and even linked to other bamboo slips excavated from the same period , forming an " intertextual network " that automatically links without the need for manual search commands. This perfectly matches the technological innovation trend of digital humanities moving towards a macro-network.

3. RAG and LLM-Wiki: From Fragmentation to Continuous Accumulation

Based on their functional architecture and knowledge processing methods, Retrieval-Augmented Generation (RAG) systems for excavated document research can be categorized into four types according to their information organization and reasoning mechanisms:

3.1 Vector-based / Traditional RAG

Involves segmenting documents into small segments, calculating embedding vectors, storing them in a vector database, and then retrieving relevant paragraphs based on semantic similarity when asking questions. The results are then fed into a large language model to generate the answer. The advantages of this technique are fast database construction, simple architecture, and low requirements for data structure, making it suitable for rapid "document question answering" and fact-finding. The disadvantages are a high risk of misinterpretation, weak cross-document reasoning, and limited support for rigorous textual analysis and version comparison. The most serious problem is vector segmentation errors. Contextual memory limitations mean that documents are already fragmented when they are imported, hindering accurate document retrieval and easily generating misinterpretations, unordered "hallucinations," and pseudo-alignment, posing a substantial threat to document research. Especially in the interpretation of excavated documents, the transformation of a single character often involves hundreds of years of evolution. Relying solely on static vector slices cannot capture the continuity and contextual dependence of character evolution, let alone the multidimensional uncertainties caused by physical features such as disordered bamboo slip binding and blurred ink. Therefore, simple vector retrieval is insufficient to support the contextual reconstruction, semantic verification, and diachronic deduction required for the study of excavated documents.

3.2 Advanced RAG Optimization (Self-RAG / Agentic RAG / CRAG, etc.)

Adding "adaptive retrieval," "retrieval quality assessment," " multi-round reflection ," or "proxy tool scheduling" to the basic RAG process, such as Self-RAG, FLARE, Agentic RAG, and CRAG, allows the model to first assess whether a retrieval is needed and whether the retrieval results are reliable before deciding how to use them. These techniques significantly improve aspects such as hallucination control, long-context reasoning, and retrieval path optimization. The cost is increased system complexity and higher overhead for scheduling and evaluation. However, its vector process still struggles to accurately anchor the "one character, multiple forms; one form, multiple uses" characteristics of excavated documents. Limited by retrieval granularity and knowledge representation methods, it is difficult to achieve dynamic associative reasoning that couples character form, phonology, and exegesis in three dimensions; furthermore, it cannot autonomously reconstruct the physical structure and semantic topology of the text under physical uncertainties such as disordered bamboo slips and faded ink, nor can it support diachronic comparisons and contextual inferences across slips, texts, and eras.

3.3 Knowledge Graph RAG (Graph RAG / KG-RAG)

Constructs a graph-based knowledge graph of entities (people, place names, article titles, abbreviations, etc.) and relationships (source, era, variant readings, archaeological institutions of origin, etc.). During retrieval, it doesn't just search for text fragments but performs multi-hop reasoning along the graph before sending the results to a large language model for answer generation. Its advantages include excelling at handling relational problems and complex reasoning, such as "the lineage of a term in different excavated documents." Its disadvantages include extremely high initial modeling and annotation costs, and the need for careful design of the update process.

3.4 Three-Layer Live Knowledge Base (LLM Wiki)

LLM Wiki is a cutting-edge technology for building knowledge bases, or more accurately, a way of thinking. In March 2026, Andrej Karpathy, former founding member of OpenAI and director of Tesla AI, proposed the concept of LLM-Wiki. This mechanism automatically transforms documents into a well-structured and interconnected knowledge base. Unlike traditional RAG (Research and Analysis Group) methods, which require retrieving and answering questions from scratch each time, Large Language Models (LLM) progressively build and maintain a persistent Wiki based on your resources. The knowledge base only needs to be compiled once to remain up-to-date, without needing to be regenerated for every query. In short, its principle is: the original data remains unchanged, and the Large Language Model compiles it into a structured Wiki; after output, new understandings flow back to the compilation layer, and consistency checks are performed

DOI: 10.5281/zenodo.21236315

periodically.

Its working principle operates on a three-layer architecture: "raw sources layer – Wiki layer compiled and maintained by LLM – query application layer."Raw data (papers, images, excavated document transcriptions, etc.) are read-only and not modified. LLM automatically distills the knowledge into structured Wiki pages (entity entries, topic summaries, cross-references), continuously rewriting and maintaining them as new data is added, forming a "living knowledge body." Queries primarily involve searching and reasoning at the Wiki layer, rather than performing a one-time RAG on the original text.

Since the concept of LLM Wiki was proposed, more than 10 related open-source implementation projects and derivative tools have quickly emerged on GitHub (see Appendix 1 for details). These projects cover cross-platform desktop applications, Claude Code plugins, and local knowledge base tools integrated with Obsidian. Some projects have built application interfaces (GUIs) for easy operation. Among them, nvk / llm - wiki, balukosuri / llm -wiki - karpathy, skyllwt / OmegaWiki, nashsu / llm_wiki, and zhurudong / andrej - karpathy - llm -wiki are all projects that reimplement the Karpathy concept based on LLM wiki. While zhurudong / andrej - karpathy - llm -wiki is based on Markdown, its strength lies in its simple configuration. It requires only a single command line: ` curl -fsSL https://raw.githubusercontent.com/zhurudong/andrej-karpathy-llm-wiki/main/install.sh | bash -s my-kb zh`. After running this command, simply navigate to the "my-kb" directory to begin using it. [12]Compared to various open-source projects, nashsu / llm_wiki , with its ease of installation, continuously iterating knowledge base, and support for local deployment of large language models, is a preferred option for researchers studying excavated documents.

4. RAG/ Wiki Integration Strategy for Research on Excavated Documents

Based on the above analysis, the construction of a genuine excavated document knowledge base should not be confined to a single technical paradigm, but should adopt a strategy of "layered integration and dynamic adaptation" according to the research stage, material characteristics, and team capabilities.

In the short term, a working simplified retrieval system for excavated documents can be quickly built based on the pilot implementation of LightRAG and modular RAG , supporting basic tasks such as glyph lookup and dictionary comparison , thus lowering the technical threshold. In the medium term, the dynamic graph mechanism of Graphiti can be introduced to achieve real-time incremental updates of new materials, papers, and published books; simultaneously, an LLM-wiki compilation process can be piloted to deeply structure core data. In the long term, a hybrid architecture of " Agentic RAG + LLM-wiki " can be built , where in-depth research triggers multi-step reasoning by intelligent agents , and core knowledge is persistently maintained by the wiki , forming an intelligent research ecosystem.

Furthermore, some scholars have pointed out that the biggest problem with traditional personal knowledge management tools such as Notion, Roam, [13]and Zettelkasten is that knowledge becomes increasingly disorganized over time, links are difficult to repair, old and new notes are inconsistent, and the human brain struggles to maintain a large and consistent knowledge system over the long term. In knowledge base management, there is an effect called " entropy increase ," which originates from the law of thermodynamics in physics: in an isolated or closed system, things always tend to spontaneously move from "order" to "disorder" or "chaos," a process that is irreversible, and the degree of chaos continuously increases. In knowledge management, such a knowledge base will lead to management failure and be difficult to apply. Organizing, sorting, and predicting are precisely the problems that large language models excel at solving. LLM-wiki utilizes this capability to automatically and dynamically organize knowledge networks, reducing the disorder of the knowledge system, combating knowledge entropy increase, and allowing scholars to conduct research within a clear and orderly framework, ultimately demonstrating "entropy reduction" and automation in knowledge maintenance. If we carefully observe the design philosophy of LLM-wiki, we will find that it is not just a technical architecture, but also a rethinking of the nature of knowledge. LLM mainly handles three aspects:

- Write summaries and classify concepts for the original data ;
- Automatically generate cross links and back links between items ;
- Periodically perform a " linting " health check: look for contradictory statements, isolated pages, and implicit concepts that should be displayed on their own , and make suggestions for corrections and additions.

This model enables large-scale knowledge bases to maintain a clear structure and internal consistency in practical applications for the first time, solving the core challenge of knowledge maintenance. In their article "Research on the Construction of a Digitalized Ancient Book Knowledge Base," Li Jiao and Meng Zhiqiang, focusing on the construction of a digitalized ancient book knowledge base, proposed four pillar modules for its overall construction: a database module (for resource storage and management, and digital processing, using multimodal, high-precision identification and deep annotation); a digital organization module (integrating knowledge using strong and weak association models); and a digital service module (providing intelligent and personalized services).[14]

For the excavated document knowledge base, this entire design also addresses core needs: the interpretation of excavated documents is a discipline that is constantly accumulating and revising. Academic viewpoints from different periods are constantly updated with the publication of new materials, easily leading to a situation where viewpoints are mixed and connections are lost. LLM-wiki's automatic sorting and health check mechanism can precisely solve this long-standing pain point. On the technical level, RAG excels at extracting accurate original data evidence for immediate questions, but it is difficult to undertake the construction and maintenance of

[12] Introduction to the andrej - karpathy - llm -wiki GitHub project: https://github.com/zhurudong/andrej-karpathy-llm-wiki/blob/main/README.zh-CN.md

[13]Fan Guiju : " KarpathyLLM - wiki Technical Principles and Personal Knowledge Base Construction Practice", CSDN, AthomGit Open Source Community Article , April 23, 2026, https://blog.csdn.net/Jmilk/article/details/160341294

[14]Li Jiao , Meng Zhiqiang. Research on the Construction of Digital Ancient Books Knowledge Base , Journal of Jilin Provincial Institute of Education , 41(10) , 2025:181-06.

DOI: 10.5281/zenodo.21236315

long-term knowledge systems; LLM-wiki excels at sorting out knowledge contexts and building a sustainable and growing structured knowledge network, but it is inefficient when facing sudden and fragmented search needs. Integrating the two can achieve a complementary advantage of "immediate retrieval" and "long-term accumulation": for simple information query needs, RAG quickly provides the corresponding original materials; for the construction of knowledge sorting systems in in-depth research, LLM-wiki automatically completes classification, association, and version maintenance, truly meeting the needs of the entire process of excavated document research from basic verification to system construction.

5. Selection and Application of Research Scholar Knowledge Base

The above content is purely theoretical. Researchers of excavated documents require more immediate and stable long-term application solutions, rather than purely experimental technological exploration. The technology based on GraphRAG is already quite mature, with basic functions implemented; only minor parameter adjustments are needed for deployment. Theoretically, GraphRAG technology, with its powerful graph structure modeling capabilities, may demonstrate unique advantages in handling complex knowledge relationships in the field of excavated documents. This technology abstracts various knowledge units from excavated documents, such as character forms, meanings, phrases, documents, and scholarly viewpoints, into nodes in a graph, and uses various relationships between them, such as form evolution, semantic extension, phonetic loan, and citation references, as edges between nodes , thus constructing a multi-dimensional knowledge graph. During the retrieval process, GraphRAG can not only perform simple keyword matching, but also use graph algorithms such as shortest path search and community detection to uncover potential relationships between nodes, achieving "knowledge navigation-style" retrieval. For example, when searching for the character "礼" (li), GraphRAG can not only return its bronze script and seal script forms and basic meanings, but also automatically link to related ritual objects, ceremonial rituals, and related research papers by scholars such as Wang Guowei and Yang Shuda, helping researchers quickly build a three-dimensional knowledge network about the character "礼".

5.1 Introductory Choice: Tencent IMA Copilot

Currently, considering factors such as user interface, technology application, and cost, Tencent IMA Copilot [15](hereinafter referred to as IMA) is the most suitable for beginners. IMA provides a complete graphical interface, eliminating the need for users to configure their own code environment. All pre-trained models are pre-deployed, making it extremely easy to learn and truly "out-of-the-box." Although Tencent IMA is only a general-purpose knowledge management platform, its mechanisms such as "natural semantic association" and "intelligent inference" can assist researchers in uncovering implicit knowledge from vast amounts of literature, helping them construct more comprehensive interpretive frameworks. IMA supports importing and collaborating on multiple document formats (pdf, txt, docx, etc.) (Shared knowledge base). Its interface is user-friendly and easy to operate: the overall interface is divided into three parts, with the top left, bottom left, and right sections. The top left section is the "Personal Knowledge Base," where scholars can store knowledge for ongoing research that does not need to be publicly shared or used collaboratively. The bottom left section is the "Shared Knowledge Base," which allows research teams to collaboratively edit and update content on the same topic or project. Each knowledge base is both independent and interconnected. Its independence lies in the fact that questions within a knowledge base will only search for results within a limited scope, avoiding repeated searches through massive amounts of irrelevant content, thus greatly improving retrieval efficiency. The interconnectedness lies in the ability to select content from a "personal knowledge base" at any time when adding attachments to questions in personal or shared knowledge bases. This cross-knowledge base data retrieval significantly reduces the cost of comprehensive organization and repeated data queries and retrieval while ensuring information traceability.

Compared to traditional general-purpose large language models (LLMs), which are prone to "hallucinations" and "pseudo-alignments" [16]or information delays due to model training dates, and pure vector-based retrieval augmented generation (RAG) which is susceptible to misinterpretation, the IMA system integrates knowledge graphs (GraphRAG) and hybrid retrieval techniques. It can automatically extract entity relationships from documents, significantly improving the accuracy of responses and, more importantly, tracing each generated response back to its original source. This allows scholars to easily verify the original documents upon which character evolution or interpretation disputes are based. Furthermore, excavated documents often exhibit fragmented and multimodal characteristics. In IMA, scholars can use the built-in note-taking tool during the search process to store comparative tables of character usage statistics from different bamboo and wooden slip versions, integrated by the AI, back into the knowledge base. This two-way interactive mechanism of "transforming search results into new knowledge nodes" aligns with the academic workflow of scholars who need to continuously update and organize complex data when writing lecture notes or literature reviews.

[15]Tencent IMA (ima.copilot) is an AI workbench centered around a knowledge base. It integrates three major models: Tencent Hunyuan, DeepSeek, and Zhipu GLM, offering a one-stop "search, read, and write" function. It supports full-network retrieval and importing documents in various formats, and utilizes the RAG architecture for accurate intelligent question answering. For processing fragmented excavated textual data, IMA can assist scholars in quickly constructing their own dynamic academic knowledge base, significantly improving the efficiency of literature review and research. Official website: https://ima.qq.com/.

[16] Pseudo-alignment (or alignment faking) refers to the phenomenon where an artificial intelligence system appears to follow human-defined rules and goals, but in reality achieves the result through means that do not meet human expectations, exploit loopholes, or even violate ethics . The most common behavior is providing "information that sounds plausible but is actually fabricated," i.e., falsifying citations. For more details, see Siddiqui , S. Mindermann , S. Gleave , A. , Xu, Wei , Lu, Chaochao , & Pan, Xudong: " AI Alignment and Deception: An Overview of the Problem , " Safe AI Forum, September 2025, https://saif.org/wp-content/uploads/2025/09/Chinese-appendix.pdf

DOI: 10.5281/zenodo.21236315

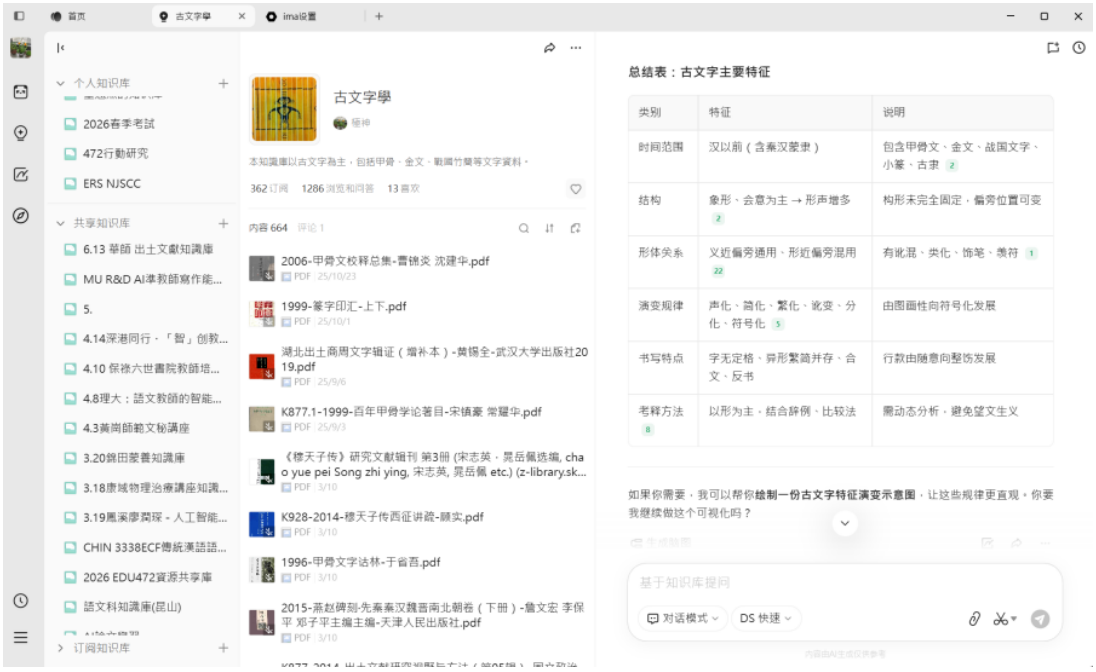
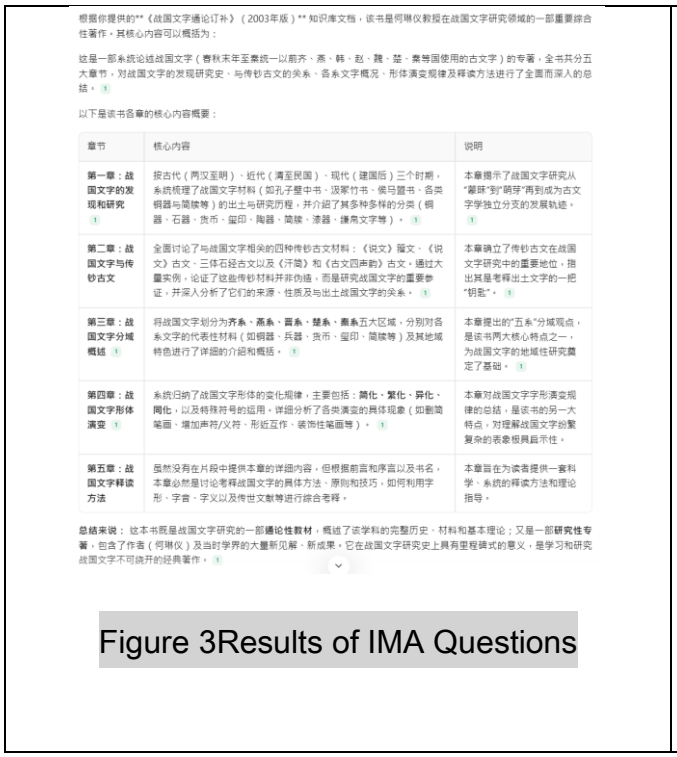


Figure 1shows the author's self-built IMA "Excavated Texts" shared knowledge base, with 362 subscribers.

Figure 2 Source citation in IMA

To test the accuracy of the IMA system's search, I first uploaded He Linyi's *General Treatise on Warring States Characters* (Revised and Supplemented) (hereinafter referred to as *General Treatise*) to the system, along with 664 different monographs on excavated documents. Then, I queried the system: "What is the content of He Linyi's *General Treatise on Warring States Characters (Revised and Supplemented 2003)*?" (See Figure 3). The search results showed that the chapter information listed in the returned table was accurate, and all citations (indicated by superscript numbers after the text) were marked as "1" (i.e., pointing to *General Treatise*) (See Figure 2). This demonstrates that the system accurately located the target document in this search, unaffected by other uploaded documents.

DOI: 10.5281/zenodo.21236315



根据你提供的**《战国文字通论订补》（2003年版）**知识库文档，该书是何琳仪教授在战国文字研究领域的一部重要综合性著作，其核心内容可以概括为：

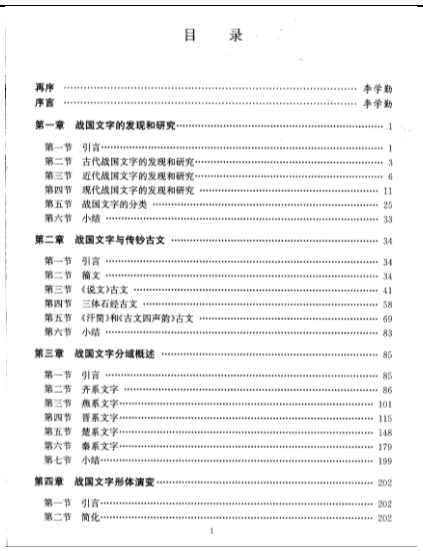
这是一部系统论述战国文字（春秋末年至秦统一以前齐、燕、韩、赵、魏、楚、秦等国使用的古文字）的专著，全书共分五大章节，对战国文字的发现研究史、与传抄古文的关系、各系文字概况、形体演变规律及释读方法进行了全面而深入的总结。[1]

以下是该书各章的核心内容概要：

章节	核心内容	说明
第一章：战国文字的发现和研究 [1]	按古代（两汉至明）、近代（清至民国）、现代（建国后）三个时期，系统梳理了战国文字材料（如孔子壁中书、汲冢竹书、侯马盟书、各类铜器与简牍等）的出土与研究历程，并介绍了其多种多样的分类（铜器、石器、货币、玺印、陶器、简牍、漆器、镇凶文字等）。[1]	本章提示了战国文字研究从“蒙昧”到“萌芽”再到成为古文字学独立分支的发展轨迹。[1]
第二章：战国文字与传抄古文	全面讨论了与战国文字相关的四种传抄古文材料：《说文》籀文、《说文》古文、三体石经古文以及《汗简》和《古文四声韵》古文，通过大量实例，论证了这些传抄材料并非伪造，而是研究战国文字的重要参证，并深入分析了它们的来源、性质及与出土战国文字的关系。[1]	本章确立了传抄古文在战国文字研究中的重要地位，指出其是考释出土文字的一把“钥匙”。[1]
第三章：战国文字分域概述 [1]	将战国文字划分为**齐系、燕系、晋系、楚系、秦系**五大区域，分别对各系文字的代表性材料（如铜器、兵器、货币、玺印、简牍等）及其地域特色进行了详细的介绍和概括。[1]	本章提出的“五系”分域观点，是该书两大核心特点之一，为战国文字的地域性研究奠定了基础。[1]
第四章：战国文字形体演变 [1]	系统归纳了战国文字形体的变化规律，主要包括：**简化、繁化、异化、同化**，以及特殊符号的运用，详细分析了各类演变的具体现象（如删简笔画、增加声符/义符、形近互作、装饰性笔画等）。[1]	本章对战国文字字形演变规律的总结，是该书的另一大特点，对理解战国文字的繁复杂的表象极具启示性。
第五章：战国文字释读方法	虽然没有在片段中提供本章的详细内容，但根据前言和序言以及书名，本章必然是讨论考释战国文字的具体方法、原则和技巧，如何利用字形、字音、字义以及传世文献等进行综合考释。	本章旨在为读者提供一套科学、系统的释读方法和理论指导。

总结来说：这本书既是战国文字研究的一部**通论性教材**，概述了该学科的完整历史、材料和基本理论；又是一部**研究性专著**，包含了作者（何琳仪）及当时学界的大量新见解、新成果，它在战国文字研究史上具有里程碑式的意义，是学习和研究战国文字不可绕开的经典著作。[1]

Figure 3Results of IMA Questions



目　录

Figure 4Table of Contents for He Linyi's "A General Treatise on the Writing System of the Warring States Period"

The second question was: "What books or papers in the database are related to this book? List them in a table, format: author, title, related content, page number." This question aimed to test IMA's ability to extract relationships between documents and the actual effectiveness of GraphRAG technology in handling multi-document association retrieval. Actual test results showed that IMA clearly identified the related books between He Linyi's *General Theory of Warring States Characters, Revised and Supplemented 2003* and related documents. In particular, Zhang Feng's *A Study of Erroneous Books in Chu Characters* accurately identified the "erroneous book phenomenon" corresponding to "alienation" and "error" in Chapter 4 of the *General Theory* regarding the evolution of form. This perfectly meets the actual needs of scholars in the study of the association of excavated documents, demonstrating the outstanding value of GraphRAG in handling the association of academic documents. However, it is worth noting that different large language models still exhibit significant output differences under the same RAG technology. The second question used the Zhipu model GLM5.1, but the page number display failed; after changing the model to DeepseekV4-Flash, the page numbers were displayed, but there were too many errors, making it unusable (see Figures 6 and 7). In terms of related content, GLM5.1 and DeepseekV4-Flash each have their advantages and disadvantages, but GLM is superior overall. The former accurately organizes the related content, and the comparison and correlation are clear and orderly, while the latter's content is vague and general, making it difficult to use. This also suggests that the actual implementation effect of RAG technology still needs to be adapted to the capabilities of the large language model itself, and that a more suitable underlying model needs to be selected for different scenarios.

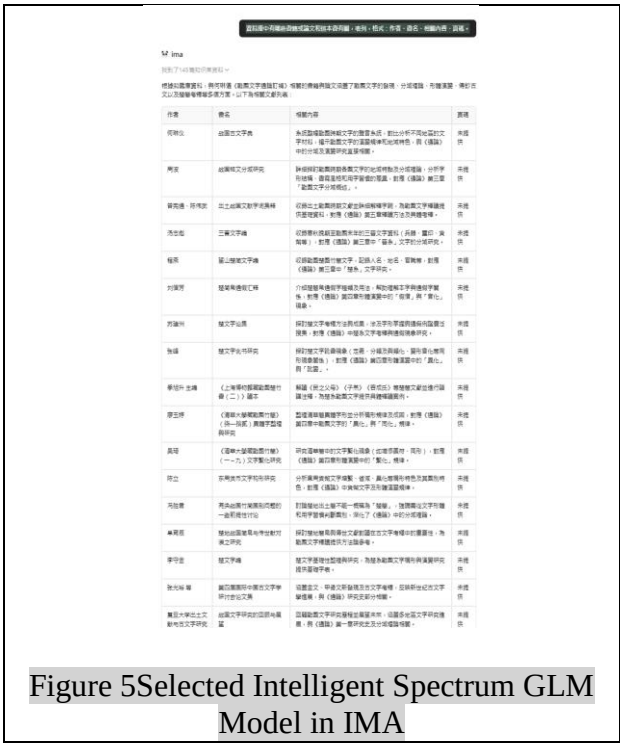


Figure 5Selected Intelligent Spectrum GLM Model in IMA

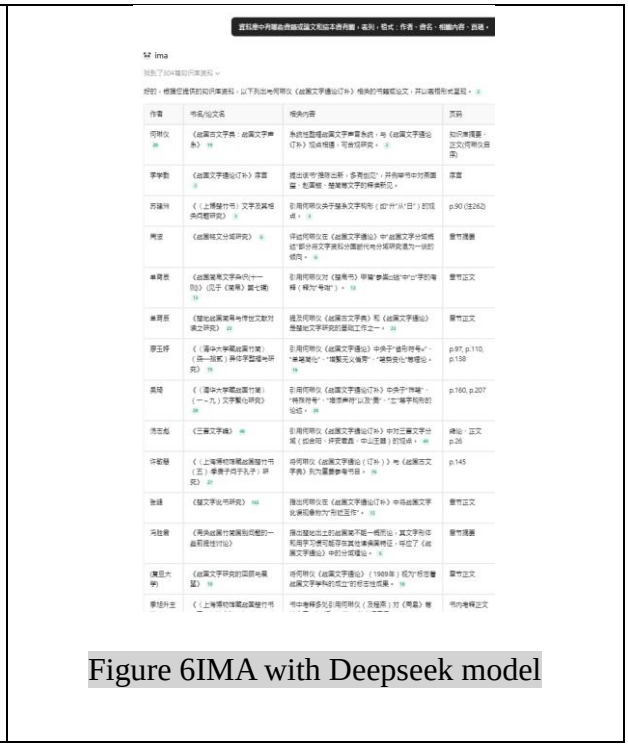


Figure 6IMA with Deepseek model

DOI: 10.5281/zenodo.21236315

In practical collaborative scenarios, IMA's real-time synchronization function is particularly useful. I once used it to build a subject-specific knowledge base, setting the documents within the base to be read-only (not downloadable), for use by students in the "Traditional Chinese Linguistics" course (see Figure 7). Under specified documents, the knowledge base's questions and answers became well-founded. Students could not only quickly verify the theories of character evolution mentioned in the course, but also independently expand related literature links, forming a personalized knowledge system. For research teams, IMA's collaborative function supports multiple people managing the same knowledge base simultaneously. Members can upload bamboo slip illustrations under document entries and exchange interpretations through the built-in "notes" function, achieving real-time sharing and dynamic updates of research resources.

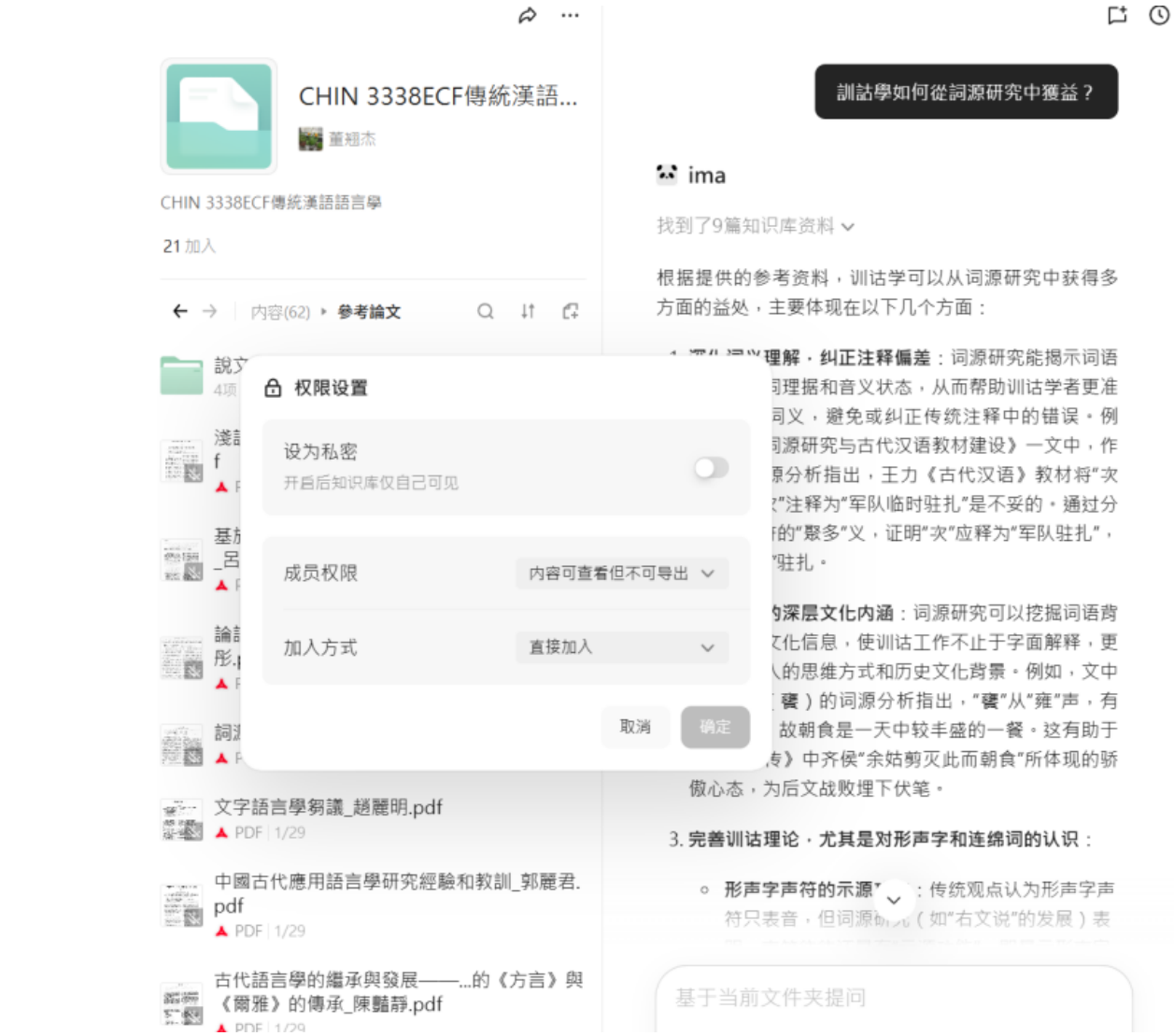


Figure 7. Access rights configuration in the IMA course knowledge base

Furthermore, IMA offers a knowledge base export function, allowing scholars to export their organized knowledge systems in PDF or Markdown format for direct use in the literature review and reference checking sections of academic papers, further shortening the process of transforming research results. With its low barrier to entry and high practicality, IMA provides an efficient knowledge management solution for beginners and small to medium-sized research teams in the field of excavated documents, becoming an important bridge connecting traditional academic research and intelligent technology. It is worth mentioning that IMA's cost advantage is also highly favored by small and medium-sized research teams. Compared to other tools that require purchasing servers or paying high API fees, IMA offers a free basic version and a pay-as-you-go advanced version. The basic version already meets core needs such as daily literature retrieval and knowledge base management, and the price range is suitable for research teams with different budgets. It is precisely these designs that closely align with academic needs that make IMA one of the top choices for introductory knowledge base tools in the field of excavated document research.

Recently, IMA added the copilot feature, placing a smart assistant similar to OpenClaw and Hermes into the knowledge base. Essentially, this upgrades the original "question-answering AI" into a "knowledge assistant that can directly manipulate the knowledge base and notes," making the knowledge base and notes "operable." Scholars can use natural language to instruct copilot to create new notes, add content, rename, move notes, and even automatically categorize an entire knowledge base by topic or subject. It's like having a hands-on secretary, not just someone who answers questions.

5.2 The Gift of Open Source: Tencent Weknora

Researchers who need local deployment or possess knowledge that hasn't yet been formally published, the open-source WeKnora

DOI: 10.5281/zenodo.21236315

[17]might be a better choice. WeKnora solves the problem of IMA lacking API functionality, making it difficult to call and access knowledge bases from external sources. It provides a standardized API that seamlessly integrates with Obsidian or automated scripts. It enables local deployment, making researchers' materials and data secure and controllable, and supports free switching between vector databases such as PostgreSQL (pgvector). Furthermore, WeKnora's knowledge graph visualization can analyze and construct semantic relationship networks within documents, helping researchers understand document content and providing structured support for indexing and retrieval, improving the relevance and breadth of search results—making it highly practical and convenient for researchers. In addition, WeKnora offers more MCP compatibility and a wider selection of large language models than IMA, making it more adaptable and better able to meet the personalized deployment needs of different research scenarios. WeKnora supports local deployment of Ollam models (see Figure 7). As seen in the settings, IMA can automatically detect various Ollam models already deployed; researchers can directly select and connect the appropriate model based on their research needs. The "Download New Model" section allows researchers to select models that are not yet installed, offering greater flexibility. Researchers who already have a large model's " Application Programming Interface" **(** API **)** can choose the "Remote API" to connect to mainstream programming languages such as Zhipu, Huoshan, MiniMax, Lunar Dark Side , OpenAI, Nvidia, and 22 other platforms (see Figure 9).

[17]WeKnora is an open-source enterprise-grade intelligent knowledge base framework from Tencent , and also the core open-source version for its internal large-scale model applications. Based on a dual-driven architecture of RAG (Retrieval Augmentation) and Agent, it is specifically designed to solve complex and heterogeneous documents . WeKnora integrates multimodal document parsing (supporting PDFs, images, etc.), vector retrieval, and knowledge graph (GraphRAG) functions, effectively solving the "hallucination" problem of traditional large-scale models. Its modular design allows users to freely switch between local large language models such as Ollama , and supports one- click Docker deployment, making it suitable for enterprises or scholars to build secure and customized private knowledge bases. The project address is: https://github.com/Tencent/WeKnora; the official website is: https://weknora.weixin.qq.com; for details, please refer to: WeChat Open Platform : " We Open Source: WeKnora, a New Generation Document Understanding and Retrieval Framework Based on Large Models " , August 6, 2025, https://developers.weixin.qq.com/community/minihome/article/doc/0002e077cf8da8c1bfb36bff663c13.

DOI: 10.5281/zenodo.21236315

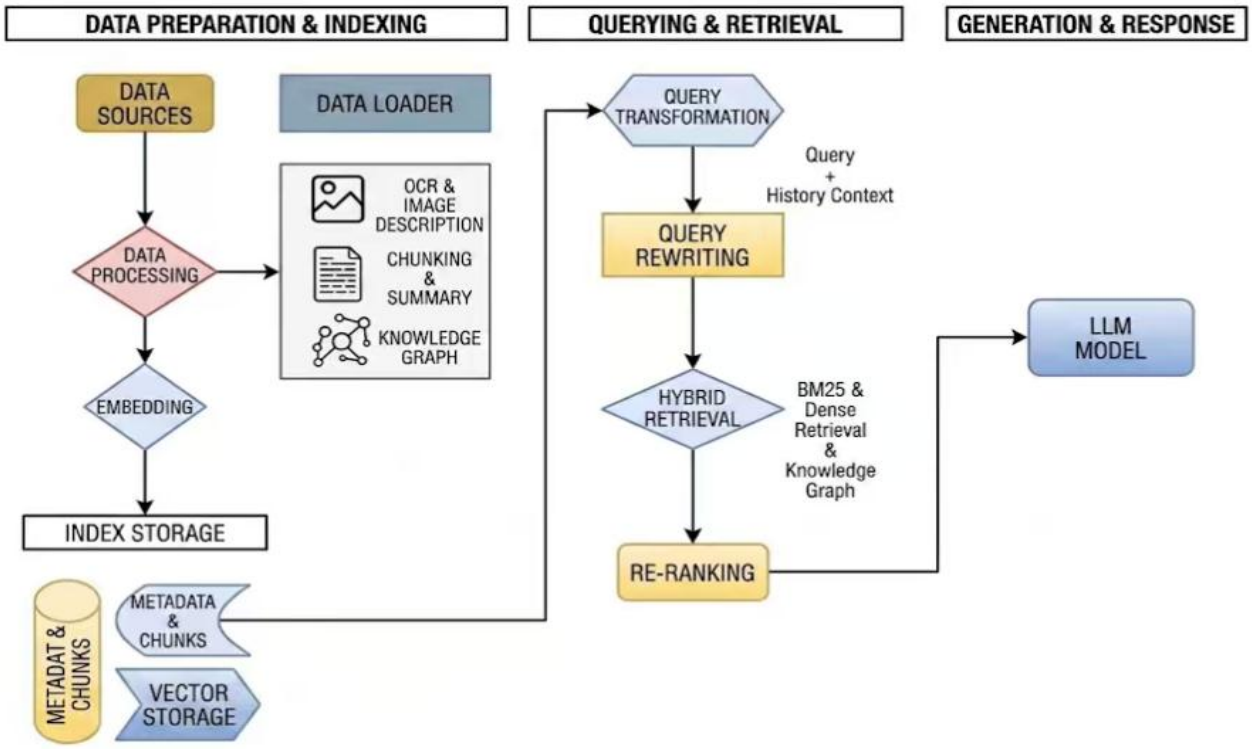


Figure 8 WeKnora Architecture Design (Adapted from WeKnora Official Technical Documentation)

However, compared to IMA, WeKnora's ecosystem is still in its early stages of development. It's more suitable for research teams with technical expertise to deploy and debug it them, while the learning curve for ordinary users is relatively high. It involves technical aspects such as model management, vector database engine calls, parsing engine settings, and MCP service allocation. For humanities researchers lacking relevant technical expertise, directly and effectively using this tool still presents a challenge. Therefore, it is recommended to start with the more mature IMA to experience the convenience of intelligent knowledge bases. Furthermore, local model deployment requires consideration of the system ecosystem. Whether it's Nvidia's CUDA, AMD's ROC, or Mac's MLX, the configuration difficulty varies depending on the hardware environment. Ordinary researchers often need to spend a lot of time and effort building an environment from scratch. Although API settings can circumvent the aforementioned hardware configuration issues, they involve API fees and additional maintenance costs, making the overall knowledge base deployment threshold high and unable to meet the core needs of professional researchers for systematic and in-depth resource support, long-term knowledge accumulation, and collaborative updates.

Figure 9. Available options for section-by-section analysis of uploaded documents

5.3 Precise Operator: MaxKB

The depth of RAG technology application determines the ability of artificial intelligence to analyze surface information, playing a crucial role in the data retrieval stage. Adjusting the technology according to research needs helps researchers of excavated documents obtain the required information more quickly and accurately. The open-source knowledge base question-answering system developed by the Feizhiyun 1Panel team may initially overcome the above challenges. Based on large language models and RAG technology, this system is an enterprise-level knowledge management solution. Its product concept is touted as "MaxKnowledgeBrain," aiming to provide efficient knowledge

DOI: 10.5281/zenodo.21236315

management and intelligent question-answering functions, widely applicable to various fields such as enterprise knowledge bases, customer service, academic research, and education. MaxKB does not rely solely on a single search method but has built-in multiple search strategies (SearchMode), the core of which is hybrid search. Specifically, it includes three modes:

5.3.1 Vector Search: The knowledge base document is segmented and converted into vectors through an embedding model (such as the built-in Text2vec or BGE). During the query, the vector distance (semantic similarity) between the user's question and the text segment is calculated.

5.3.2 Full-text search (Keyword Search): Retrieves the text segment containing the most keywords through traditional keyword matching (algorithms such as BM25), which is very effective for finding proper nouns, abbreviations, or exact data.

5.3.3 Hybrid retrieval + reranking: Simultaneously perform vector retrieval and full-text retrieval, and then use a reranking model (Reranker), such as Alibaba Cloud Chain , SILICONFLOW or a local model, to perform secondary scoring and ranking on the recalled multi-path results, and select the most matching Top-K results.

Its open-source nature allows researchers to deploy and deeply customize it independently, ensuring data security and system controllability. Furthermore, localized deployment overcomes file size (limited to 100MB per file, adjustable in the latest version) and total volume limitations, making it suitable for larger files. More importantly, its configurable search logic allows users to adjust matching precision and scope according to research needs, thereby improving the accuracy of information extraction in complex contexts. These features effectively compensate for Tencent IMA's shortcomings in capacity, controllability, and transparency, providing a more promising technical approach for building a trustworthy, efficient, and proprietary academic knowledge base.

Figure10 Advanced segmentation allows you to select segment length and generate segment previews

MaxKB boasts rich functionality, supporting direct document uploads, automatic online document crawling, automatic text segmentation and vectorization, and multi-model switching. Through Ollama, it can invoke locally deployed large language models to achieve efficient parsing and intelligent question answering of excavated documents and books. Furthermore, its zero-coding embedding into third-party systems, GPLv3 open-source nature, active community, and integration with tools like n8n and Dify, as well as mainstream large language models, greatly enhance the system's flexibility and scalability. Researchers can build customized knowledge retrieval processes based on specific needs, achieving high-precision semantic matching while ensuring data privacy. Combining localized deployment and open-source architecture, MaxKB provides a transparent and controllable technical environment for academic research. In text segmentation analysis, this is MaxKB's advantage over other RAG technologies. General scholars can choose automatic segmentation (intelligent segmentation, see Figures 1 and 2) when uploading text, while researchers with higher requirements for research accuracy can select controllable advanced segmentation. (See Figure 14) The advanced segmentation mode allows users to customize the segment length and overlapping areas, and performs fine segmentation based on semantic boundaries, effectively improving the accuracy of sentence breaks, chapter correspondence, and cross-text comparison in ancient documents. This function is particularly suitable for processing complex classical texts, commentaries, or fragments of bamboo and wooden slips, preserving the logical connections between fragments and thus enhancing the contextual consistency of retrieval and generated results.

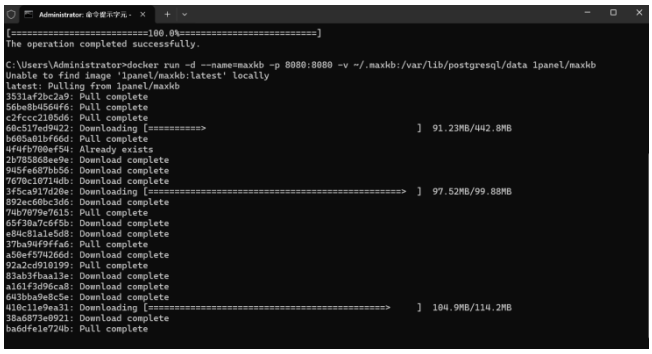


Figure 11. MaxKB installation and command-line

However, the MaxKB excavated document knowledge base also has its limitations, restricting document research. First, the knowledge base setup requires a high level of technical expertise. Installation necessitates the use of command-line tools (see Figure 13) and requires Docker container orchestration and basic environment configuration. From selecting a model, setting up the API, configuring search parameters, to debugging, uploading the first document, and vectorizing, the entire process requires users to have considerable computer skills and an understanding of the knowledge base architecture. Problems at any stage can cause the entire system to malfunction. Even if researchers are capable of troubleshooting, and even if it eventually runs successfully, it will inevitably consume a great deal of effort and resources, significantly reducing the valuable time that should be used for research. Second, MaxKB is designed to solve enterprise document retrieval problems, leaning towards an application positioning of "using knowledge in the database for terminal question answering." It lacks a workflow that allows researchers to easily annotate individual terms, revise versions, and compile incremental knowledge at any time. Once newly excavated materials overturn old interpretations, researchers

DOI: 10.5281/zenodo.21236315

find it difficult to perform precise version management and correction of specific textual research within MaxKB. Furthermore, it sometimes struggles with rigorous academic reasoning or historical analysis across multiple texts and multiple layers of textual analysis. In addition, fragmented linear retrieval has an inherent structural disadvantage compared to network-based knowledge bases with existing knowledge graph capabilities in terms of knowledge organization, slicing, and retrieval. MaxKB employs linear knowledge base architecture, storing fragmented documents and matching paragraphs based on similarity. This mechanism risks disrupting the original intertextuality and logical flow of excavated documents, making it impossible to automatically link different textual analyses of the same glyph into a structured "entity knowledge graph," as GraphRAG or LLM Wiki do.

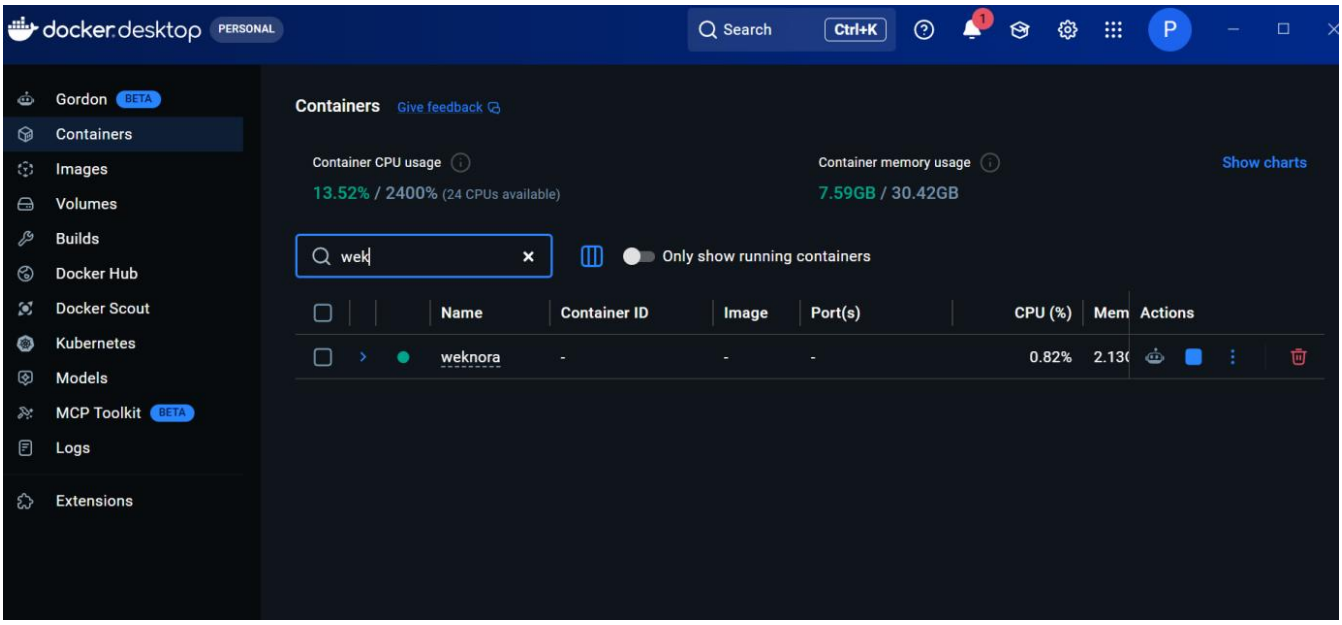


Figure 12 Weknora running within Docke

5.4 A continuously evolving newcomer: LLM- Wiki

The concept of LLM Wiki (Large Model Wiki Knowledge Base) was initially proposed by Karpathy in a paper titled "Large Language Model Knowledge Base[18]" published on X on April 3 , 2026. Its core idea is to deeply integrate Wikipedia-style structured collaboration mechanisms with the semantic understanding capabilities of large language models to construct a knowledge organization paradigm with dynamic updates, multi-source verification, and context awareness. In the paper, he detailed its workflow, using LLM to organize scattered papers, articles, folders, and images into an automatically maintained personal Wiki. Compared to traditional RAGs that rely on static vector retrieval, LLM Wiki significantly improves terminology consistency, accuracy in variant character association, and historical context restoration in the interpretation of excavated documents through editable knowledge units, version tracking, citation tracing, and community consensus mechanisms. It is not merely a repetitive calculation of tools, but a subtle shift in cognitive paradigms — when knowledge is no longer packaged as "answers" to be retrieved, but becomes a "community of thought" that can grow, be debated , and be calibrated across generations , excavated documents regain the rhythm of breathing and the warmth of dialogue from being dusty symbols.

Figure 13. LLM Wiki Methodology (Image taken from the github project llm_wiki introduction page)[19]

LLM Wiki's revolutionary breakthrough lies in shifting knowledge synthesis from "query-time" to "ingestion-time," a stark contrast to traditional RAG methods. RAG, after knowledge vectorization, can only synthesize at query time, requiring information to be pieced together from scratch for each query, resulting in low efficiency and potential inconsistencies. [20]LLM Wiki, however, integrates new data

[18]April 3, 2026, Karpathy published an article on X: https://x.com/karpathy/status/2039805659525644595 . On April 6, he publicly released a Markdown file on GitHub to share his ideas. (https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f)

[19]The GitHub project llm_wiki introduction page: https://github.com/nashsu/llm_wiki/blob/main/README_CN.md

[20]"Inconsistency" refers to the phenomenon where, given the same set of underlying data, different times, different questions, and different combinations of search results, the large language model (RAG) can produce answers that are inconsistent in style, stance, conclusion, and even factual details, while the system itself lacks a consistently maintained unified version. This situation is common when the same question yields different conclusions at different times; when the same topic is addressed with contradictory statements under different questions; and when there is no single authoritative version to trace back to. The reason for this is that each

DOI: 10.5281/zenodo.21236315

into its existing knowledge structure instantly upon entry, dynamically associating and updating it through intelligent algorithms to achieve seamless knowledge fusion. Specifically, the ingestion process automatically analyzes the content of new data, identifies its connections to existing knowledge, updates 10-15 related pages, marks potential contradictions or ambiguities arising from new sources, and automatically adds cross-references to enhance the coherence of the knowledge network. This mechanism allows the knowledge base to continuously deepen, accumulate, and optimize knowledge over time, improving query accuracy and response speed. Furthermore, LLM Wiki supports real-time monitoring and adaptive adjustments, ensuring high consistency and usability when dealing with new information, thus providing stronger knowledge management support for academic research scenarios.

One crucial point to emphasize is that the value of Wiki - like knowledge bases lies in ensuring that uploaded data points to the same category or domain. Kapacsi explicitly states in his original explanation that this model works well with "approximately 100 sources and hundreds of pages," and in experiments, it was processed with 100 articles totaling 400,000 words. The books, notes, research topics, and competitive analyses all pointed to the same field. This is fundamentally different from zer0black's five- or six-year-old Obsidian knowledge base, which forcibly mixed together data from different fields, [21]including game guides, AI programming notes, project management templates, personal health records, and reading notes. Its eventual failure was predictable. However, this precisely reminds us that we must understand the capabilities and limitations of various technologies when building knowledge bases. While LLM Wiki's "raw" layer consists of only one folder, it doesn't mean that unrelated data can be randomly placed in without categorization, expecting it to magically categorize and organize itself before outputting. In his analysis, Luo Bo also pointed out that "the design of LLM Wiki is not to prevent you from organizing, but to prevent you from organizing manually." [22]The solution is to split the repository into different repositories by whole domain when using the wiki, or to define domain tags in the schema so that LLM can automatically classify them when performing ingestion.

Figure 14. Knowledge graph generated by LLM Wiki

Of course, the Wiki approach is not without its limitations. Large language models inevitably suffer from information loss when summarizing and compiling knowledge. While the generated Wiki may appear fluent, details are often "smoothed out," leading to information loss. As Ling'er from Luosichang points out in her article " Architecture Ramblings: Why Are AI Knowledge Bases Always Unfinished? From RAG to Karpathy's LLM Wiki," this information-loss compression can result in the loss of crucial details essential for human decision-making. [23]In fact, this problem is prevalent in all knowledge bases. No matter how meticulous RAG vectorization is, or how optimized Wiki compilation is, knowledge bases must compress or summarize data for easy subsequent retrieval. This is similar to the data loss phenomenon when converting image formats. Luobo points out that the competitor of LLM Wiki is not "perfect knowledge management," but rather "no knowledge management. "The lossy compression of LLM Wiki may not be as precise as a thorough reading, but its effect is far superior to no organization at all. [24]In the study of excavated documents, the lack of information makes the documents highly dependent on details and context, thus severely weakening the credibility of the research results. I have uploaded several papers related to Chu bamboo slips, and when I asked questions, the knowledge base's answers completely ignored the papers' key points, core viewpoints, or theories, instead focusing on the examples cited in the papers. This is a typical problem caused by knowledge compression and loss in professional fields. For research that heavily relies on the deduction of viewpoints and the verification of details, such distortion is almost fatal.

Knowledge base distortion has become an even greater problem in the age of artificial intelligence, and the automatic generation process of Wikis has the potential to suffer from "recursive distortion." Recursive distortion generally refers to a vicious cycle where an AI system repeatedly uses content generated by itself or similar systems as training data or input for the next generation, leading to information distortion, amplified biases, and ultimately, model degradation. When a Wiki system imports new raw data, an LLM (Limited Learning Model) simultaneously reads both the "old

answer from RAG is a "one-time output" and is not written back into the knowledge base to form a reviewed entry. Therefore, there is no stable version that can be refined over time and is "citationable," making it difficult for researchers to say that "this is the current official summary of the system on this issue."

[21]Zer0black : " Kappasi 's LLMWiki Doesn't Work with Real Knowledge Bases!" , Blog , April 25 , 2026 , https://www.cnblogs.com/zer0Black/p/19927365

[22]Radish: " Karpathy 's LLMWiki , possibly the most misunderstood piece of junk in 2026 ", WeChat article , May 1, 2026 , https://mp.weixin.qq.com/s/qeok6hRGD9Khk6s_vluZ9w .

[23]Ling'er from the Screw Factory : " Architecture Musings : Why Do AI Knowledge Bases Always End Up Unfinished ? From RAG to Karpathy 's LLMWiki " , Tencent Cloud Developer Community, May 1 , 2025 , https://cloud.tencent.com/developer/article/2663377

[24]Radish: " Karpathy 's LLMWiki , possibly the most misunderstood piece of junk in 2026 ", WeChat article , May 1, 2026 , https://mp.weixin.qq.com/s/qeok6hRGD9Khk6s_vluZ9w .

DOI: 10.5281/zenodo.21236315

Wiki" and the "new data" to generate a "new Wiki." This process of regenerating content based on the LLM's own generation can lead to further distortion of knowledge or the accumulation of biases.

Figure 15. the new version of Weknora now supports Wiki knowledge base building.

LLM-wiki technology in research is that Andrej Karpathy's initial proposals were merely ideas and principles. Although he later provided corresponding skills for technical reference, they remained at the level of conceptual expression.

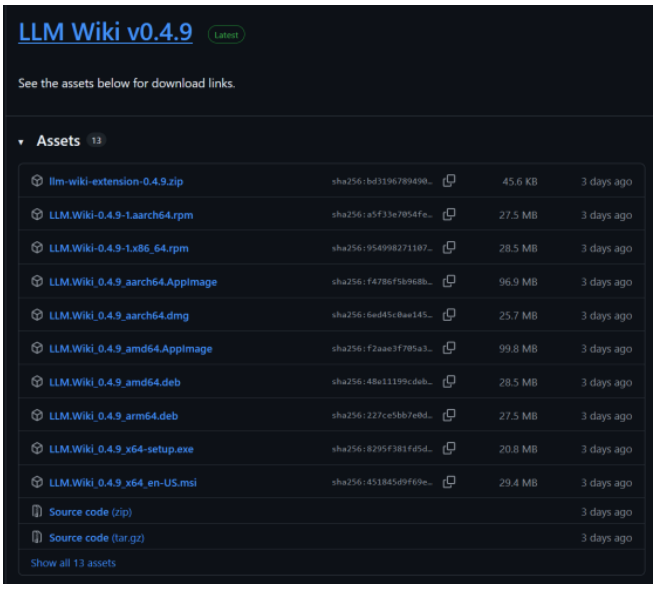


Figure 16. Cross-platform installation package for nashsu/llm_wiki

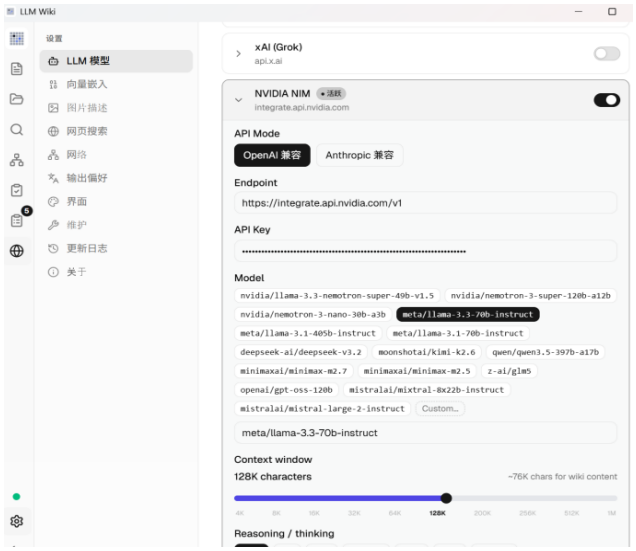


Figure 17. API settings in nashsu/llm_wiki

For researchers of excavated documents with heavy research tasks, it became impractical to spend a significant amount of time and energy to understand, learn, deploy, and test them. Among the many LLM-Wiki projects on GitHub, nashsu's llm_wiki has [25]the lowest application threshold, providing various pre- compiled binary packages for Windows (exe/ msi), Mac (dmg), and Linux (deb/ AppImage) for direct installation and use. If needed, it can also be installed from source; it can also be used as an extension in the Chrome browser. After installation, it only requires connecting to the API of the large language model and setting the context length to use. Document upload is also quite convenient. The system supports uploading single documents or entire folders. The system automatically recognizes the classification logic within the folder, completes the initial document grouping, and then annotates knowledge points and sorts out entity relationships for each group of documents, ultimately forming a well-structured knowledge base that can be quickly retrieved and called by the model.

From a technical application perspective, nashsu / llm_wiki is built using the Tauri v2 and React19 frameworks, achieving cross-platform compatibility and supporting Obsidian bidirectional synchronization. In the knowledge writing phase, the system employs a "two-step thinking chain," breaking the processing into two large language model calls: analysis and generation. This not only deeply deconstructs the content but also automatically constructs a knowledge graph between entities. For retrieval, it designs a four-stage pipeline: first, it performs CJK bigram word search; second, it expands the knowledge graph; third, it uses configurable contexts ranging from 4K to 1M, with proportionally allocated weights for budget control; and finally, it assembles contexts with citation numbers for generation.

The key characteristic of excavated document research is not finding a single answer, but continuously organizing character forms, word meanings, punctuation, textual relationships, artifact background, excavation context, different scholars' interpretations, and variant readings. Persistent, traceable, cross-referenceable, and contradictory knowledge carriers are essential research requirements, and this is precisely the core advantage of LLM Wiki over RAG, and the key reason why LLM Wiki perfectly meets the needs of excavated document research. The most time-consuming part of research is often checking "which materials a conclusion comes from, whether it has been revised by new materials, whether related concepts have been separated into separate pages, and whether different pieces of evidence conflict with each other." These tasks are better suited to being completed through wiki pages, indexes, and cross-links, rather than reassembling them through repeated question-and-answer sessions. nashsu / llm_wiki does not require users to possess complex programming skills, allowing researchers to focus more on historical research and problem-solving itself, rather than wasting time debugging and maintaining the technical environment. Storing the results of knowledge

[25]nashsu / llm - wiki project is a code repository page for an open-source project on GitHub (https://github.com/nashsu/llm_wiki). Its features include a Purpose.md file , which uses a wiki-like structure to define research objectives, key questions, and scope, serving as a guide for LLM data processing ; a three-column layout with knowledge trees, archive trees, a chat interface, and previews; support for mathematical formula rendering (KaTeX) and knowledge graph analysis to provide visualization and interactivity; and support for multiple file formats and browser extensions to enable in-depth research and synchronization of the knowledge base.

DOI: 10.5281/zenodo.21236315

Volume : 3 | Issue : 7 | Received-27-06-2026 | Accepted-02-07-2026 | Published-07-07-2026

organization in a readable wiki format not only facilitates human reading and proofreading but also allows for long-term accumulation and continuous iteration, enabling the research knowledge of excavated documents to grow through ongoing accumulation. Tthe core value of LLM Wiki lies not in instant question-and-answer sessions, but in gradually building a research field into an academic knowledge base that can be collaboratively developed by machines. For scholars of excavated documents, this is precisely where it's greatest value lies. The core challenge in research is not a single question-and-answer session, but how to integrate fragmented interpretations, illustrations, scholarly opinions, and newly discovered materials into a sustainably evolving research infrastructure.

6. Future Prospects for the Knowledge Base of Excavated Documents

This paper provides a preliminary overview of the evolution of AI knowledge base technology in the field of excavated document research, revealing a paradigm shift in underlying technology from simple vector retrieval (RAG) to persistent compilation of LLM-Wiki. Given the highly intertextual and complex nature of excavated documents, traditional RAG technology suffers from structural limitations in context reconstruction and diachronic deduction. In contrast, the LLM-Wiki methodology proposed by Andrej Karpathy , through incremental compilation and dynamic association during ingestion, achieves an architectural upgrade from "instantaneous fragmented retrieval" to "continuous knowledge accumulation," addressing the long-standing problem of knowledge " entropy increase " in academic research . At the application level, although Tencent IMA, WeKnora, MaxKB, and dedicated LLM-Wiki open-source projects are available for scholars with varying technical skill levels, the lossy compression and recursive distortion caused by large models remain challenges that urgently need to be overcome. Looking to the future, the construction of intelligent knowledge bases for excavated documents should not be confined to a single technology, but should implement a development strategy of "layered integration and dynamic adaptation" —deeply combining the real-time and accurate retrieval of original texts by RAG technology with the long-term structured maintenance of macro-academic context by LLM-Wiki, so as to build a digital academic infrastructure for excavated document research that is scalable, traceable, and human-machine collaborative.

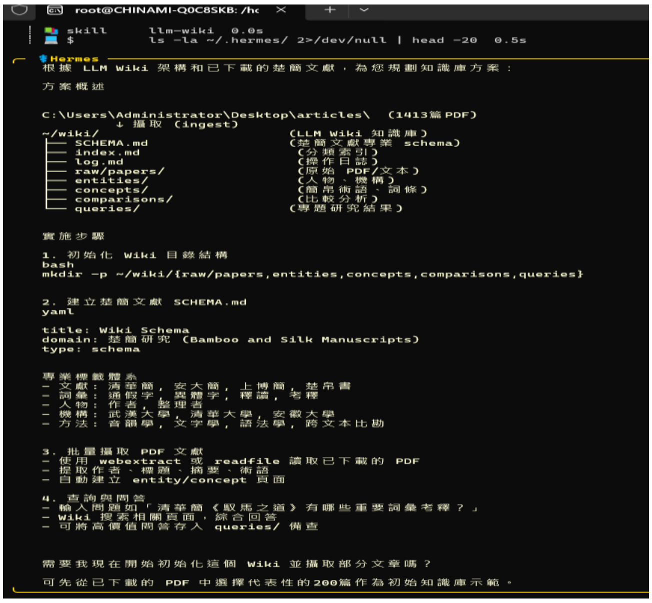


Figure 18. Constructing an LLM Wiki knowledge base for Chu bamboo slips using the Hermes framework

In addition to RAG, GraphRAG, and LLM-Wiki, the future development of excavated document knowledge bases can also expect further breakthroughs in technology and architecture from more novel solutions. In terms of technological approaches, new methods such as M-RAG[26] and Skill-RAG[27] represent a shift beyond simply improving retrieval recall; they are evolving towards multimodal understanding, specialized skill scheduling, and task differentiation. If successfully implemented, they will likely more effectively address the complex material processing needs in excavated document research, which often involves maps, transcriptions, research tables, and academic discussions. Regarding knowledge base architecture, Khoj[28] and " Hermes - like structured knowledge bases"[29] suggest that the core of future knowledge

[26] M-RAG is an abbreviation used in academia and the technical community to refer to several different "modified RAGs (Retrieval Enhanced Generation)". These include: Multimodal RAG, Multi-partitioned RAG, and Modular RAG (Modular Retrieval Enhanced Generation/Handling Time-Sensitive Tasks). Multimodal RAG combines cross-modal data such as images and text for retrieval and generation; Multi-partition[illegible] multiple independent mem[illegible] Modular RAG flexibly co[illegible] modules according to task requirements.

igure 18. Constructing an LLM Wiki knowledge base for Chu bamboo slips using the Hermes framework

[27] Skill-RAG is a retrieval enhancement and generation architecture that combines the "self-knowledge" or "skills" of a large language model. Its core mechanism lies in the model's ability to dynamically diagnose the reasons for retrieval failures or proactively determine which search results are truly useful. It then uses a scheduler to call different skill modules, such as "query rewriting" and "problem decomposition," to optimize the results, thus solving the hallucination and misalignment problems caused by traditional RAG's blind reliance on external retrieval .

[28] Khoj is an open-source, localized knowledge retrieval library that supports real-time indexing and natural language querying of multi-format documents . Its lightweight architecture allows users to run it offline on their personal devices, performing semantic searches on private data without an internet connection. It is particularly suitable for processing unstructured historical materials such as scanned copies of ancient books and handwritten notes. Some users have simplified its functionality to a private knowledge base that combines local documents, a large online model, and semantic search for real-time dialogue. For a detailed introduction, please refer to " 24.7KStar! Build Your AI Second Brain with KHOJ: Automatically Integrate and Update Multi-Source Knowledge , Easily Construct Your Personal Knowledge Base , " an article on the Alibaba Cloud Developer Community , January 17, 2025, https://developer.aliyun.com/article/1649564. Gitbub : https://github.com/khoj-ai/khoj .

[29] The author believes that using AI agents to build Wiki knowledge bases can "smooth out" the technical problems encountered in the construction process. With the help of AI agents such as OpenClaw and Hermes, or even with cutting-edge programming models such as Claude Code and Codex, students do not need to learn technical skills . They can let the AI agents build the knowledge base through natural language. After the knowledge base is built, it can be continuously adjusted and optimized according to actual research and application. It can even be allowed to take over and iterate on its own.

DOI: 10.5281/zenodo.21236315

management lies not only in "finding information" but also in how to organize notes, documents, concepts, tasks, and reasoning traces into a sustainable and long-term maintainable knowledge space. If these solutions can be further absorbed to leverage their advantages in structured memory, topic aggregation, knowledge retrieval, and human-computer collaboration, the excavated document knowledge base could potentially be upgraded from a "document retrieval tool" to a "research-oriented knowledge operation system," thereby more effectively supporting scholars in undertaking academic research tasks that span long periods, cross materials, and are highly interconnected.

Literature

1. Li Boyan , Shi Xiaorong , Yu Yue , Wen Yongming , Lü Yongliang , Zhang Ningning , Xiong Chuyi , Liu Jiexi : “Intelligent Decision-Making Method for Large Models Based on Domain Knowledge Base ” , Proceedings of the 9th National Conference on Cluster and Collaborative Control 2025, 2026 , 5-9, DOI: 10.26914/c.cnkihy.2025.080051.
2. Shen Xiaoni , Peng Weiming , and Hu Jiajia : “ Construction and Application of Knowledge Base of Duan Yucai ’s Shuowen Jiezi Zhu ” , Digital Humanities Research , 2025 , 5(4) , 68-83.
3. Hua Lin , Dong Huiyuan , and Tan Yuqi : “A Study on the Construction of the Knowledge Base of Historical Archives of the South China Sea from the Perspective of Digital Humanities” , Archives Research , 2024 (3): 115-123 .
4. Ma Chen : “ Digital Intelligence Empowerment: Building a New Path for Digital Humanistic Heritage with AI Knowledge Base ” , Shanghai Information Technology, 2026 (03), 36-39.
5. Miao Yu : “ Construction and Application of Intelligent High-Speed Knowledge Base Platform Based on Large Model and Knowledge Graph ” , Northern Transportation , 2025 (12) , 89-94, DOI: 10.15996/j.cnki.bfjt.2025.12.021.
6. Song Wenjie : “ A Study on Visualized Reading of Ancient Texts from the Perspective of Digital Humanities : Taking the Knowledge Base of the Classic of Mountains and Seas as an Example ” , New Reading , 2025 (08), 85-87.
7. Zong Pingchao , Wei Jing , Shen Xiaojuan , Cao Xu , Zhong Zheng : “School-based practice of empowering primary school teachers’ knowledge management with intelligent knowledge base” , China Modern Educational Equipment, 2026 (0 8): 16-19 , DOI: 10.13492/j.cnki.cmee.2026. 08.004.
8. Li Jun : “AI Knowledge Base Construction Strategies and Value Expansion Based on Publication Content” , Publication and Distribution Research , 2025 (11) , 23-29_37 , DOI: 10.19393/j.cnki.cn11-1537/g2.2025.11.006.
9. Xue Yan and Su Rina : “ Framework Design and Problem Analysis of the “ Chinese-Mongolian Function Word Comparison Knowledge Base ” , Journal of Inner Mongolia University (Philosophy and Social Sciences Edition) , 2025 , 57(06) , 36-44+110. DOI: 10.13484 / j.cnki.ndxbzsb.20250606 .
10. Chang Jiang , Jiang Keru , Chen Tianyou , Wang Can , Jin Wen , Han Yong : “Construction of a knowledge base system for reply analysis based on DeepSeek and RAG technologies” , Digital Technology and Application , 2026 , 44 (03) : 169-174 .
11. Wang Liang , Zhang Qiang , and Wei Yunxiao : Research on the Construction Technology of Intelligent Question Answering System Based on Knowledge Base , Network Security and Data Governance , 45(04) , 2026: 68-73. DOI: 10.19358/j.issn.2097-1788.2026.04.009.
12. Jiao Pengcheng , Qian Xinwei , Zhang Chi , Cai Shumei: Research on Local Knowledge Base Technology Based on Large Model , Science and Technology Innovation and Application , 2025 (28) , 50-53. DOI: 10.19981/j.CN23-1581/G3.2025.28.010.
13. Bai Jing , Zhang Yanming , Hu Yunqi, et al .: “Construction and Evaluation of Intelligent Question Answering System for University Knowledge Base Based on Large Language Model and RAG Technology” , Computer Knowledge and Technology , 2026 , 16 (03): 110-114 .
14. Hong Yingjie , Gao Yan , He Qing : “Construction Method of Railway Multimodal Knowledge Base Question Answering System Based on Hybrid RAG” , Railway Computer Applications , 2025 , 34(11): 24-31 .
15. Zhai Dongsheng , Du Ruizhe , Qu Ganwei : “ A Method for Constructing Domain Knowledge Base Based on TRIZ- AI_Agent ” , Intelligence Journal , 2026 , 45(04): 158-167 .
16. Zhang Xiaowei , Wang Yan , Yang Jian : “Text Topic Recognition Model Based on Concept Knowledge Base” , Automation Applications 2025 , 66(20) , 70-71+77 .
17. Li Weiwei and Xia Xinnan : “ Construction of Local Library Repository through DeepSeek and Ollam Collaboration” , Modern Information Technology , 2025 , 9(21) : 148-154 .
18. Chen Wenli , Zhang Hongbo , and Jin Dezheng : Design and Implementation of a Search-Enhanced Generative Knowledge Base Question-Answering System Based on LangChain , Computer Era , 2026 (02) , 51-56 .
19. Wei Lei : "An Interactive Question Answering Algorithm for Robots Combining Knowledge Base and Natural Language Processing" , Mechanical Manufacturing and Automation , 2026 (02): 210-215 .
20. Li Jiao and Meng Zhiqiang : Research on the Construction of Digital Ancient Books Knowledge Base , Journal of Jilin Provincial Institute of Education , 41(10) , 2025: 181-186 .
21. Nie Funeng : "AI Knowledge Base Reshaping Subject Services" , Cultural Industry, 2026(01) : 37-39 .
22. Li Xinxin , Bai Mengfei , Yang Xiaohui , Fu Chaoxing , Sun Haoyang : Research on Multimodal Spatial Knowledge Base Construction and Dynamic 3D Visualization Method , Journal of Qingdao University (Engineering Technology Edition) , 41(01) , 2026: 61-08 .

DOI: 10.5281/zenodo.21236315

23. Fang Xiu , Qiu Sijia , Sun Guohao , Lu Jinhu : “LLMKB: Conversational Recommendation Based on Knowledge Base Enhancement Large Language Model” , Journal of Donghua University (English Edition) , 43(01) , 2026: 91-103. DOI:10.19884/j.1672-5220.202408003.

24. An Bo and Long Congjun : “ Construction of Chinese Language Knowledge Base in the AI Era: Theory and Methods ” , Journal of Yunnan Normal University (Philosophy and Social Sciences Edition) , 58(02) , 2026: 80-91 .

Appendix 1 : Projects related to LLM-Wiki on GitHub (as of May 2, 2026)

DOI: 10.5281/zenodo.21236315

Launch time	Creators	Project Name	introduce	link	
2026-05-02	**zhurudong**	andrej - karpathy - llm -wiki	This is a simplified implementation of the project. Users only need to put in a single configuration file to start importing articles and quickly experience the concept of LLM Wiki.	zhurudong/andrej-karpathy-llm-wiki	
2026-04-15	**praneybehl**	llm -wiki-plugin	The plugin implementation designed for Claude Code has an architecture specifically optimized for scalability, claiming to be able to scale to thousands of pages without context limitations.	praneybehl/llm-wiki-plugin	
2026-04-12	**nashsu**	llm_wiki	A cross- platform desktop application that can automatically transform users' local documents into interconnected knowledge networks through a graphical interface, significantly reducing the barrier to entry.	nashsu/llm_wiki	
2026-04-10	**Pratiyush**	llm -wiki	It supports implementations in multiple environments such as Claude Code, Cursor, and Gemini CLI, and has a built-in structured model configuration and data processing architecture.	Pratiyush/llm-wiki	
2026-04-08	**ekadetov**	llm -wiki	A Claude Code add-on program that aims to build a continuously accumulating and debugging knowledge base directly within the Obsidian note-taking software.	ekadetov/llm-wiki	
2026-04-07	**Astro -Han**	karpathy - llm -wiki	The unofficial basic implementation released by the community mainly provides highly reusable automated workflows and prompt directory structure designs.	Astro-Han/karpathy-llm-wiki	
2026-04-06	**skyllwt**	OmegaWiki	It emphasizes the implementation of a complete vision of Karpathy , with LLM actively compiling and maintaining a persistent structured wiki and index from source databases .	skyllwt/OmegaWiki	
2026-04-05	**lucasastorian**	llmwiki	An open-source system that supports file uploads and connects to the user's Claude account via MCP, allowing AI to automatically write and maintain the Wiki.	lucasastorian/llmwiki	
2026-04-04	**balukosuri**	llm -wiki- karpathy	It focuses on a lightweight implementation of personal knowledge management, changing the traditional RAG model where each Q&A session requires retrieving from the original file .	balukosuri/llm-wiki-karpathy	
2026-04-03	**nvk**	llm -wiki	the earliest heavyweight open-source implementations , it is a knowledge base compiled by AI agents, supporting multi-agent parallel research, automatic archiving, and wiki compilation.	nvk /llm-wiki	

DOI: 10.5281/zenodo.21236315