AIGC-Empowered Efficient Personalized Lesson Preparation System

ChenXi Nan
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1364734102@qq.com

Abstract. To address the increasing demand for personalized instruction in education and ameliorate the time-consuming and inflexible nature of traditional lesson planning, this paper presents the development and evaluation of an intelligent, AI-based lesson preparation tool. The system integrates Large Language Models (LLMs), including the Langchain-Chatchat framework, ChatGLM3, and ERNIE-3.5-8K, with the objective of significantly enhancing both the efficiency of teacher preparation and the quality of instruction. The tool implements four core functionalities: rapid question-answering based on a local knowledge base, one-click intelligent generation of instructional images and PowerPoint presentations (PPTs), and automated assignment generation and grading. Results from functional testing and user feedback evaluations indicate that the tool operates stably, effectively reduces educators' preparation time, enriches the dimensions of teaching content, and increases student engagement. The successful implementation of this research provides a valid paradigm for the deep application of artificial intelligence technology in the education sector, demonstrating its substantial potential in advancing personalized learning and automating pedagogical tasks.

Keywords: Educational Technology； Artificial Intelligence；Intelligent Teaching Resources；Teaching Efficiency

1. Introduction

As pedagogical approaches continue to diversify—evolving from traditional textbook-based instruction to online multimedia learning, and from monolithic lecture formats to interactive discussion and inquiry-based exploration—the content and structure of educators' lesson preparation have become increasingly complex. Conventional lesson planning, primarily relying on textbook review, reference material consultation, and manual authoring of lesson plans, exhibits significant limitations. This traditional process demands substantial investment of educators' time and effort in information retrieval, organization, and plan composition. The task of identifying and curating high-quality, course-appropriate resources is particularly time-consuming, frequently consuming the majority of preparation time.

The rapid advancement of Artificial Intelligence (AI) has introduced disruptive transformations in this domain, demonstrating immense potential to address the shortcomings of traditional teaching and lesson planning methodologies. This project focuses on the development of an intelligent lesson planning tool designed to offer a more efficient and streamlined preparation process for educators. By integrating advanced models such as Langchain-Chatchat, ChatGLM3, and ERNIE-3.5-8K, this initiative aims to tackle the prevalent issues of low efficiency and resource scarcity in conventional lesson planning, while simultaneously exploring novel applications of AI technology within the educational field.

2. Technical Selection

2.1 Knowledge Base and Information Retrieval in Langchain-Chatchat

The Langchain-Chatchat architecture empowers developers to build end-to-end applications leveraging Large Language Models (LLM) for tasks involving local knowledge bases with private data. This framework supports a variety of data types, including unstructured files, and facilitates the integration of LLMs with supplementary computational resources or external knowledge bases to

https://doi.org/10.61957/joerai-20260105

create more efficient and powerful software solutions. It enables users to extract valuable information from extensive unstructured data, which is then utilized for subsequent queries. The process involves converting document content into vector representations, which are subsequently stored in a dedicated vector database. When a user initiates a query, the system generates a corresponding query vector and utilizes the Faiss library to efficiently retrieve the top-ranked relevant information. This retrieved information is then aggregated into a prompt template and fed to the LLM to generate a response for the user.

Building a local knowledge-based large model with the ChatGLM and Langchain-Chatchat stack not only provides the capability to process unstructured data but also offers broad applicability across various domains, including natural language processing, question-answering systems, and text generation. This approach lays the groundwork for developing more intelligent and efficient question-answering systems and provides robust support for future optimization and expansion [1]. The construction of a vector knowledge base primarily comprises two stages: text vectorization, and the storage and management of the resulting vectors.

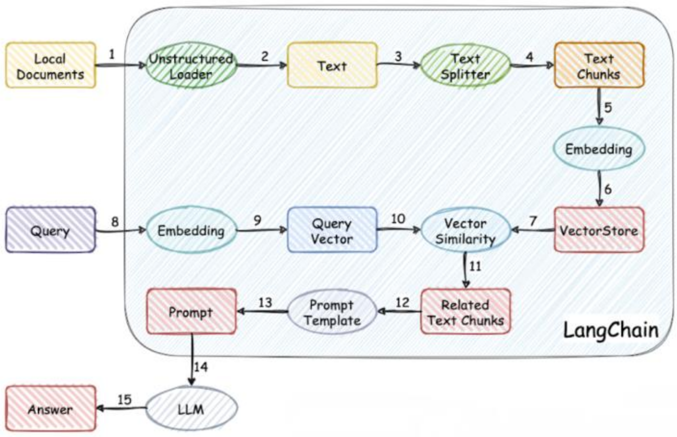

Figure 1. LangChain-Chatchat Architecture Diagram

In the process of text vectorization, an embedding model maps lexical units to numerical vectors, thereby enabling computational systems to interpret human language within a mathematical framework. This mapping is designed to preserve semantic similarity, such that words with proximate meanings are positioned closely in the vector space, while dissimilar words are located at a greater distance from one another. Through this vectorization, textual content is transformed into a format that is amenable to computational processing.

This project utilizes the m3e-base embedding model. This model was selected for its robust performance in use cases that are predominantly Chinese, with a limited presence of English.

Table 1 Comparison Results of Embedding Models

	Parameters	Embedding Dimension	Chinese	English	s2s	s2p	s2c	Open Source	Compatibility
m3e-small	24M	512	Yes	No	Yes	No	No	Yes	Excellent
m3e-base	110M	768	Yes	Yes	Yes	Yes	No	Yes	Excellent
text2vec	110M	768	Yes	No	Yes	No	No	Yes	Excellent
openai-ada-002	Unknown	1536	Yes	Yes	Yes	Yes	Yes	No	Excellent

Note. s2s (sentence-to-sentence) represents the embedding capability between homogeneous texts.
Applicable tasks: text similarity, duplicate question detection, and text classification.

s2p (sentence-to-passage) represents the embedding capability between heterogeneous texts.
Applicable tasks: text retrieval and GPT memory modules.

s2c (sentence-to-code) represents the embedding capability between natural language and programming language.
Applicable task: code retrieval.

Compatibility indicates the degree to which a model is supported across various projects in the open-source community. Since both m3e and text2vec can be directly used via sentence-transformers, their community support is comparable to that of OpenAI.

The M3E model employs in-batch negative sampling for contrastive learning, a method that ensures the effectiveness of negative sample selection. It was trained for a single epoch on a dataset comprising over 22 million sentence pairs. Specifically, M3E leverages a large-scale Chinese sentence-pair dataset that encompasses diverse domains such as encyclopedias, finance, medicine, law, news, and academia, totaling 22 million sentence pairs. Additionally, the model was trained on the MEDI dataset, which contains 1.45 million English triplet examples. To enable the M3E model to follow instructions during text encoding, a fine-tuning dataset with over 3 million instruction examples was utilized. As an all-in-one text embedding model, M3E supports not only homogeneous sentence similarity tasks but also heterogeneous text retrieval.

For the storage and management of vectors, the Faiss library is employed to handle large-scale vector data, enabling the efficient identification of the nearest neighbors to a target vector within a vast collection. A core principle of Faiss indexing is Product Quantization (PQ), which encodes points in the vector space into a finite subset. In Approximate Nearest Neighbor (ANN) search, PQ achieves vector compression by partitioning the vector space into M subspaces and applying vector quantization to each subspace. This approach significantly reduces computational complexity during queries by enabling rapid comparisons between the query vector and the subspaces of each database vector. Another key technology is the Inverted File system (IVF), which partitions the data space into multiple clusters and applies PQ to each cluster. IVF facilitates fast localization and searching across the entire dataset. The essence of IVF lies in constraining the search space to a subset of clusters proximate to the query point through a priori clustering, thereby reducing global computation and sorting operations.

By integrating an embedding model with the Faiss library, the Langchain-Chatchat framework can establish a knowledge base that is both capable of efficiently storing vast amounts of information and rapidly retrieving relevant content. Within this knowledge base, textual information is first converted into vectors by the embedding model and then stored in a specialized data structure optimized for fast nearest neighbor search. When an educator needs to retrieve information, the large model simply converts the query text into a vector and uses the Faiss library to find the most similar vectors in the knowledge base, returning the information associated with them. This method of storage and retrieval substantially improves the efficiency and user experience of the knowledge base.

2.2 The ChatGLM-6B Model

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

ChatGLM-6B is an open-source, bilingual (Chinese and English) conversational language model. Based on the General Language Model (GLM) architecture, it comprises 6.2 billion parameters.

https://doi.org/10.61957/joerai-20260105

Through the application of model quantization techniques, this model can be deployed locally on consumer-grade graphics cards, requiring a minimum of only 6GB of VRAM at the INT4 quantization level. The activation function employed by ChatGLM-6B is the Gaussian Error Linear Unit (GELU) [2].

Prevailing pre-training frameworks can be categorized into three main paradigms: auto-regressive, auto-encoding, and encoder-decoder models. While auto-regressive models have demonstrated significant success in long-text generation and exhibit few-shot learning capabilities when scaled to billions of parameters, their unidirectional attention mechanism inherently limits their capacity to fully capture the dependencies between contextual words in Natural Language Understanding (NLU) tasks. In contrast, auto-encoding models learn bidirectional context encoders through denoising objectives, yielding effective contextualized representations for NLU tasks; however, they are not directly applicable to text generation. Encoder-decoder models employ bidirectional attention for the encoder and unidirectional attention for the decoder, with cross-attention mediating between them. This architecture is typically utilized for tasks such as text summarization and response generation.

None of the auto-regressive, auto-encoding, or encoder-decoder models can concurrently achieve optimal performance across NLU, unconditional generation, and conditional generation tasks. To address these limitations, GLM was proposed as a general-purpose pre-training framework that utilizes an autoregressive blank-filling approach. GLM is trained by optimizing an autoregressive blank-infilling objective. Given an input text $x = [x_1, ..., x_n]$, multiple text spans $\{s_1, ..., s_m\}$ are sampled, where each span s_i corresponds to a sequence of contiguous tokens $[s_{i,1}, ..., s_{i,li}]$ in x. Each span is replaced by a single [MASK] token, creating a corrupted text xcorrupt. The model then predicts the missing tokens within the spans in an autoregressive manner, conditioned on the corrupted text. When predicting the missing tokens for a given span, the model has access to the corrupted text and all previously predicted spans. To fully capture the interdependencies between different spans, GLM randomly permutes the order of the spans and then generates the tokens within each blank from left to right.

The input x is partitioned into two parts: Part A, which is the corrupted text xcorrupt, and Part B, which contains the masked spans. Tokens in Part A can attend to each other but cannot attend to tokens in Part B. Tokens in Part B can attend to tokens in both Part A and the preceding parts of Part B, but not to subsequent tokens in Part B. Each span is prepended with a special [START] token for input and appended with an [END] token for output. Through this mechanism, the model implicitly learns a bidirectional encoder (for Part A) and a unidirectional decoder (for Part B) within a single, unified architecture.

2.3 The Stable Diffusion Model

Stable Diffusion is a generative model based on Latent Diffusion Models (LDMs), which learns data distributions to generate images. Diffusion models have demonstrated superior perceptual quality compared to Generative Adversarial Networks (GANs) and better density estimation than autoregressive models1. However, a significant drawback is that generating high-quality samples requires hundreds or even thousands of model evaluations.

The image generation process in LDMs begins by employing an autoencoder to compress high-dimensional image data into a lower-dimensional latent space. This approach preserves the essential perceptual information of the data while reducing computational complexity. By training and performing generation within this latent space, LDMs strike a more effective balance between computational efficiency and the quality of the generated output. To prevent excessive spatial compression and ensure that the latent space retains sufficient detail, the autoencoder is trained to provide a low-dimensional representation that is perceptually equivalent to the original data space. This methodology not only reduces computational costs but also enhances the overall efficiency of the image generation process.

Subsequently, the diffusion model is trained within this latent space. The training involves a forward process of progressively adding noise to the latent representations and a reverse process of learning to denoise them to generate new data. Traditional diffusion models operate directly in the pixel space, which incurs substantial computational overhead due to the high dimensionality of the data being

https://doi.org/10.61957/joerai-20260105

processed. In contrast, LDMs execute the diffusion process in a low-dimensional latent space, which significantly reduces the computational load at each step and thereby improves both training and inference efficiency.

A notable advantage of LDMs is that the versatile autoencoding stage needs to be trained only once; it can subsequently be reused for the training of multiple diffusion models or adapted for various tasks. This modularity allows for the efficient exploration of diffusion models across a range of image-to-image and text-to-image applications. For instance, by connecting a Transformer architecture to the U-Net backbone of the diffusion model, it becomes possible to implement arbitrary token-based conditioning mechanisms.

In practical applications, Stable Diffusion excels at unconditional image synthesis, image inpainting, and stochastic super-resolution. Furthermore, by leveraging its universal, cross-attention-based conditioning mechanism, the model can be trained for class-conditional, text-to-image, and layout-to-image synthesis.

3. System Design

3.1 System Overview

To enhance the efficiency of lesson planning for educators, this project integrates a suite of advanced artificial intelligence technologies and tool libraries, including Large Language Models (LLMs), image generation models, and automated document creation tools. By leveraging these technologies, the system provides several key pedagogical support functions, such as model-driven dialogue, image generation, PowerPoint (PPT) creation, assignment generation and grading, and flowchart synthesis. These features are designed to substantially improve the efficiency of the lesson preparation process. The core components of the system include several state-of-the-art AI models, namely Langchain-Chatchat, ChatGLM3, and ERNIE-3.5-8K. These models possess exceptional language understanding and text generation capabilities, which effectively support functionalities like conversational Q&A, as well as the creation and assessment of assignments. Furthermore, the integration of pptx-python, an automated document creation library, enables educators to rapidly generate well-designed and information-rich PPT instructional materials. Stable Diffusion serves as the primary technology for image generation, producing vivid and accurate images or illustrations tailored to specific pedagogical content, thereby enhancing the expressive power of teaching materials. The inclusion of Mermaid facilitates the quick generation of flowcharts, further enriching the structural clarity of the instructional content. The synergistic combination of these technologies alleviates the burden of lesson preparation, allowing educators to dedicate more focus to exploring innovative teaching methodologies and deepening student interaction. Although the system is built upon a foundation of artificial intelligence, a human-centric design principle has been consistently maintained through meticulous design and development. The system not only offers efficient tools and methods for lesson planning but also considers the habits and needs of its users—the educators—to ensure that the technology serves a supportive and assistive role. Figure 3-1 presents a schematic overview of the system.

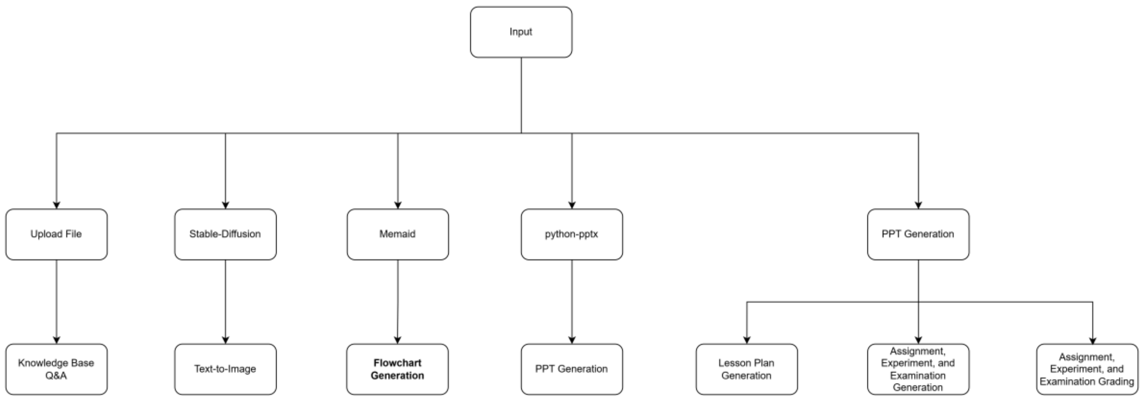

Figure 2. System Overview Diagram

https://doi.org/10.61957/joerai-20260105

In the initial phase of lesson planning, the educator uploads all relevant teaching materials to the system. During knowledge base question-answering sessions, the system automatically retrieves vectorized information to provide pertinent responses. This functionality significantly enhances the searchability and usability of educational resources, offering robust support for teachers in understanding and refining course content.

Educators can utilize this system to rapidly generate lesson plans, reducing the time required for manual authoring. Concurrently, the system can dynamically adjust the content of these plans based on students' actual learning progress. By using the automated lesson plan generation feature, teachers can allocate more time to classroom instruction. This function thus becomes a convenient and highly efficient tool in lesson preparation, contributing to an improvement in both teaching quality and student learning outcomes.

For PPT generation, the educator inputs the desired content or topic. The large model then performs a detailed expansion and optimization of the core points and key concepts. Finally, the system leverages Python to automatically generate the corresponding PPT presentation.

In the image generation module, the teacher provides a theme or content description. The large model enriches and refines this initial concept to produce a series of precise and detailed prompts. These content-aligned prompts are then fed into the Stable Diffusion model to generate relevant images.

The large model can efficiently generate a variety of assessment types, including multiple-choice questions, fill-in-the-blank questions, short-answer questions, and even complex experimental designs or comprehensive examinations. The generation of these assignments, experiments, and tests is based on a holistic consideration of learning objectives, knowledge point distribution, and student learning data. After students complete these tasks, the teacher can input their responses along with the original questions into the large model. The model then conducts a comprehensive and integrated evaluation of the students' work, moving beyond simple right-or-wrong judgments to perform an in-depth, multi-dimensional analysis.

When an educator needs to present complex information in a concise and intuitive manner, the flowchart generation function can be used to convert input text into a clear diagram. The large model extracts keywords and generates the corresponding Mermaid syntax to render the flowchart on the page.

3.2 Component Design

The Langchain-Chatchat framework integrates an embedding model with Faiss technology. The embedding model is responsible for efficiently processing documents uploaded by the educator, converting them into a vector format suitable for querying. When an educator poses a question, Langchain-Chatchat leverages Faiss to perform high-efficiency similarity searches, rapidly identifying and retrieving the most relevant information from a large volume of data. The framework accepts the raw content of the uploaded files and utilizes the m3e-base model to transform the document content into numerical vectors. During the knowledge base question-answering process, the teacher's query is similarly converted into a vector, enabling Faiss to quickly identify and extract the most pertinent content from the stored documents. Langchain-Chatchat not only significantly enhances the speed and accuracy of accessing the knowledge base but also simplifies the complex query process.

The Stable Diffusion model generates images based on input prompts. For a theme or content provided by the educator, the ERNIE model optimizes it, transforming the refined concept into a prompt compatible with the Stable Diffusion model. The model then executes the diffusion process based on these inputs to generate an image that corresponds to the provided content. This enables the system to support educators in inputting textual content based on their needs, which the model then translates into highly relevant images. The diffusion model utilized by Stable Diffusion is a type of unsupervised generative model. As Malik, the head of the open-source computer vision library OpenCV, stated in an interview, diffusion models leverage knowledge acquired from text data to understand the semantics of word combinations and connect them to the real world. This allows AI to generate more complex and varied images without relying on specific datasets[3].

https://doi.org/10.61957/joerai-20260105

Mermaid is a text-based diagramming tool that facilitates the conversion of textual content into visual charts, enabling the transformation of an educator's input into a flowchart. When a teacher inputs a block of text into the system, the large model identifies keywords and analyzes the core content. These keywords are then optimized and organized to ensure the accuracy and logical coherence of the conversion process. The large model structures these refined keywords into the Mermaid syntax format, producing a standardized string. Upon receiving and parsing this formatted string, Mermaid renders it as a structured diagram. This visualization of the textual content simplifies complex concepts and processes, providing a clearer understanding of the underlying logic and structure. By utilizing the Mermaid functionality within the project, the system allows educators to quickly convert dense or structurally complex text into intuitive and easily comprehensible flowcharts. For demonstration purposes, I have used Mermaid's official online editor to convert the required content into the appropriate syntax and render the flowchart.

The python-pptx library enables the programmatic conversion of textual content into PowerPoint (PPT) presentations. The library provides a programming interface that allows for the creation of PPT files through code, enabling the dynamic generation of slides, addition of text, adjustment of formatting, and setting of styles based on specific content requirements. After an educator inputs instructional content into the system, the large model extracts and supplements key information. The resulting educational content is then converted into a structured JSON format, which is subsequently used by the python-pptx library to create the corresponding PPT. When an educator wishes to insert images, they can first generate relevant instructional visuals using Stable Diffusion and then insert them into the PPT using python-pptx. This approach not only makes the presentations more vivid and illustrative but also enhances their visual appeal. The figure below provides a simple example of creating a PPT.

In the interface design, the left-hand side is dedicated to the question-answering section for interaction with the large model. The functional modules on the right-hand side are accessible to the teacher upon clicking. The interface layout is shown in the following figure.

4. Implementation and Practice

4.1 Knowledge Base Question Answering

Educators can upload course-related instructional materials into the system's integrated knowledge base. During the upload process, the provided documents are converted into a vector format, which facilitates subsequent semantic analysis. In the course of lesson preparation or classroom interaction, educators can pose targeted questions to the knowledge base as needed. The system then automatically retrieves the vectorized information and provides relevant responses.

A document knowledge base, beyond merely containing extensive raw document resources (such as directory databases, full-text databases, multimedia databases, and metadata databases), also integrates heterogeneous resources through processes like classification, extraction, storage, and presentation [4]. It extracts and organizes knowledge, and discovers multidimensional, network-like associations between documents through methods such as association rule mining, thereby uncovering latent knowledge through intelligent means [5].

4.2 Lesson Plan Generation

To generate a lesson plan, the educator initiates the process by providing a topic or a segment of relevant content as input. Upon receiving this input, the large model leverages natural language processing (NLP) techniques to analyze and comprehend the text. The model then extracts the core concepts, selects the material most pertinent to the pedagogical objectives, and augments this information by supplementing it with necessary background knowledge, illustrative case studies, or additional reference materials. Finally, this enriched and structured content is used to populate a pre-designed lesson plan template.

4.3 PowerPoint Presentation Generation

Based on the user-provided keywords or expanded content, the Large Language Model (LLM) first structures the material into distinct sections. For each section, it generates a subtitle, the main textual content, and a corresponding image prompt. These image prompts are then sent to the Stable Diffusion model, and the resulting images are automatically saved to a designated local directory. Subsequently, the LLM is prompted to format the textual content into a predefined JSON structure. This JSON output is parsed by a Python script, and finally, the python-pptx library programmatically generates each slide of the presentation, inserting the previously created images into their appropriate positions.

4.4 Generating Instructional Images for PowerPoint Presentations

By integrating the content summarization capabilities of ERNIE-3.5-8K with the image generation prowess of Stable Diffusion, the system enables educators to translate initial concepts or thematic outlines into custom visuals. The educator provides a preliminary idea, which the Large Language Model (LLM) then refines and expands into a series of precise and detailed prompts. These prompts are subsequently input into the Stable Diffusion model to generate customized images that are closely aligned with the instructional theme.

The resulting images can be embedded into teaching presentations or used to create engaging visual content. This process not only enhances the visual quality of the course materials but also serves to capture student attention and increase their motivation during classroom instruction.

4.5 Automated Generation and Assessment in Education

Upon receiving the instructional content from the educator, the Large Language Model (LLM) automatically generates assignments, experiments, and other relevant materials by integrating the content with the key learning objectives for the lesson. This process obviates the need for teachers to manually search for and compile problem sets. After students complete these assignments, experiments, or examinations, their responses can be input back into the model along with the original questions. The model then conducts a comprehensive, multi-dimensional evaluation of each student's work across six key areas:

(1) practical application skills

(2) mastery of theoretical knowledge

(3) proactive learning ability

(4) teamwork and communication skills

(5) innovative and critical thinking

(6) personal affective attitudes and social responsibility.

Finally, the model provides targeted feedback and improvement strategies for each student, fostering their holistic development.

As noted in existing literature, AIGC tools possess significant advantages in the context of programming education for non-computer science majors. They not only offer personalized learning support but also help to eliminate language barriers, reduce student frustration, and enable learning anytime and anywhere [6]. This automated mechanism for the design and evaluation of assignments, experiments, and examinations enhances the efficiency of the grading process for educators and also assists higher education faculty and students in learning to co-evolve curriculum reform practices with artificial intelligence [7].

4.6 Additional Features

Educators can input a segment of instructional text into the system, such as a description of a course workflow, the logical relationships between knowledge points, experimental procedures, or any other information that requires graphical representation. The Large Language Model automatically extracts the key content from the text and converts it into the Mermaid syntax format. This code is subsequently rendered by the Mermaid engine to generate and display the complete flowchart.

5. Conclusion

Contemporary reforms in the educational sector have imposed greater demands on pedagogical quality and efficiency, rendering traditional lesson planning methods insufficient to meet the needs of the modern educational environment. To address these challenges, this project integrates the ChatGLM-6B model with the Langchain-Chatchat framework to optimize the knowledge base and advance information retrieval capabilities. This integration enables the system to efficiently process vast amounts of information and deliver rapid, precise retrieval services. Furthermore, the Stable Diffusion model introduces new possibilities for content visualization by rapidly generating high-quality images directly relevant to the instructional material, thereby enriching the diversity and visual impact of classroom presentations. The system's integration of automated tools for generating and grading lesson plans and assignments, alongside a flowchart synthesis tool, significantly enhances the efficiency of lesson preparation. By leveraging these technologies, the system alleviates the preparatory burden on educators, enabling them to devote more energy to pedagogical innovation and student interaction.

References

[1] Liu, Yuehan; Huo, Haobin; Jin, Canguo.Practice and Exploration of Building an Enterprise-Level Private Large Language Model Assistant Based on ChatGLM3 and RPA Technologies. Architectural Design Management, 2023, 40(12): 33–40.

[2] Yin, Xian; Feng, Yanhong; Ye, Shigen.
Optimization Study of an Intelligent Conversational System for Aquatic Animal Disease Diagnosis Based on ChatGLM. Modern Electronic Technology [Online], 2024: 1–7 [Accessed Apr. 18, 2024]. http://kns.cnki.net/kcms/detail/61.1224.TN.20240409.1451.002.html

[3] Zhao, Juecheng.AI-Generated Art: Surprise Accompanied by Controversy. Global Times, Mar. 24, 2023 (008). DOI: 10.28378/n.cnki.nhqsb.2023.002496.

[4] Niu, Li; Han, Xiaoting.Research on the Integration and Service Models of Archival Information Resources in Cloud Computing Environments. Archives Science Study, 2013(05): 26–29.DOI: 10.16065/j.cnki.issn1002-1620.2013.05.007.

[5] Huang, Jian; Liu, Jingyi; Li, Zhe.An Analysis of the Integrated Co-Construction Model of Enterprise Document Knowledge Bases and AI Resource Pools. Zhejiang Archives, 2021(09): 53–55.DOI: 10.16033/j.cnki.33-1055/g2.2021.09.020.

[6] Bo, Junge; Qiao, Yanan; Qi, Qi; et al.Exploring the Potential and Challenges of AIGC Technologies in University Programming Courses. Computer Technology and Development [Online], 2024: 1–8 [Accessed May 28, 2024]. Available: http://kns.cnki.net/kcms/detail/61.1450.tp.20240514.1711.029.html

[7] Guo, Jianan; Zhao, Shan.Education in the Era of Generative Artificial Intelligence: Opportunities, Challenges, and Responses of ChatGPT in Promoting University Course Innovation. Educational Science Exploration, 2023, 41(06): 89–97.

https://doi.org/10.61957/joerai-20260105