INTI JOURNAL | Vol.2025, Issue 2, No.10
eISSN:2600-7320

The Application of RAG in Langchain Framework in Classical Chinese

Liu Zhi Hao[1,2], Leong Wai Yie[2*]

[1]Nantong Institute of Technology, 226000 Nantong, Jiang Su, China
[2]INTI International University, 71800 Nilai, Negeri Sembilan, Malaysia

[*]**Email:** waiyie.leong@newinti.edu.my

Abstract

Currently, the world's mainstream Large Language Models (LLMs) offer significantly less support for Chinese than for English, resulting in challenges when utilizing generative LLMs to produce high-quality Chinese traditional literature works. This paper proposes a data source creation method, this method interprets words according to their extended meanings, which means one meaning of a word produces another or several meanings related to it in the process of language development, then use a word segmentation tool to divide the different meanings of a word, which re-quantifies the nouns, verbs, stories and histories in classical Chinese, the advantage of quantifying in this way is that it can effectively solve the problem of polysemy of words, and enhances the logical correlation between contexts. From the results, the correlation between the generated classical Chinese and the real results has been greatly improved. We use the Retrieval Augmented Generation (RAG) method to get the results at the least cost without retraining the new LLM.

Keywords

RAG, Data Source Creation Method, Classical Chinese, Energy Efficiency

Introduction

Classical Chinese literature is a vital carrier of Chinese culture, characterized by its concise language and profound meaning, often presented in the form of polysemous words and allusions. (Xiang, 2024; Liu, 2022) However, there is a significant semantic gap between classical texts and modern Chinese, as many words have evolved to carry multiple meanings over time, and their grammatical structures differ significantly from those of modern Chinese. Currently, LLMs primarily use English training data, with very little Chinese corpus, especially classical literary resources.

This paper utilizes RAG technology within the Langchain framework. (Cai, 2023; Pokhrel, 2024) Langchain, as a modular development framework, provides a flexible toolchain for LLM

Submission: 23 January 2025; **Acceptance:** 15 May 2025; **Available online:** 18 August 2025

applications, supporting functions such as data retrieval, context management, and multi-step reasoning (Leong, 2025a; 2025b; 2025c). The system can first retrieve the extended meanings of target words, then guide the LLM to output text that fits the context based on dynamically generated contextual prompt words.

RAG allows us to realize our ideas without retraining new large models, instead building upon existing large models. First, retrieve the context provided by the user from the external data source, and then retrieve the relevant information to insert into the model prompt, allowing the LLM to generate the required results (Jeong, 2023). A typical RAG application has two main components (Şakar, 2025). Indexing: A process used to obtain data from a data source, generate a vector representation, and save it in a vector database. Usually done while offline; Retrieval and generation: The actual RAG chain, which receives user queries at runtime, retrieves the relevant data from the indexed data, and then passes it to the LLM.

The complete sequence from the original data to the answer comprises five parts, and the LangChain framework provides all the necessary building modules, ranging from simple to complex RAG applications: Load, Split, Store, Retrieve, and Generate. The specific process is shown in Figures 1 and 2:

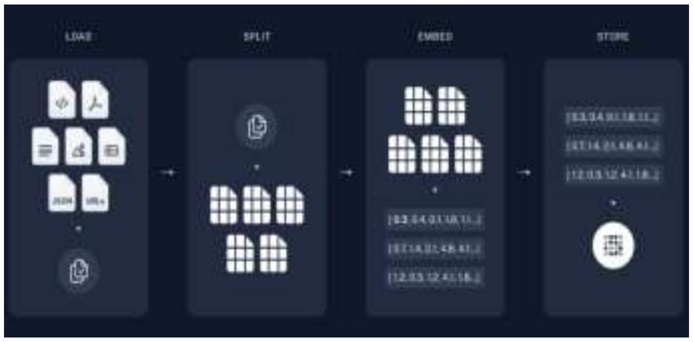


Figure 1. Load, split, and store the data sources

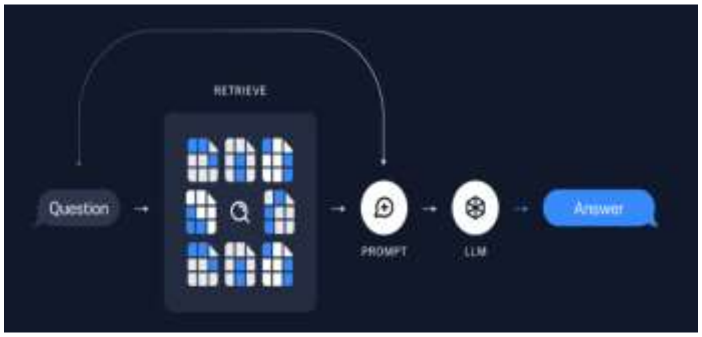


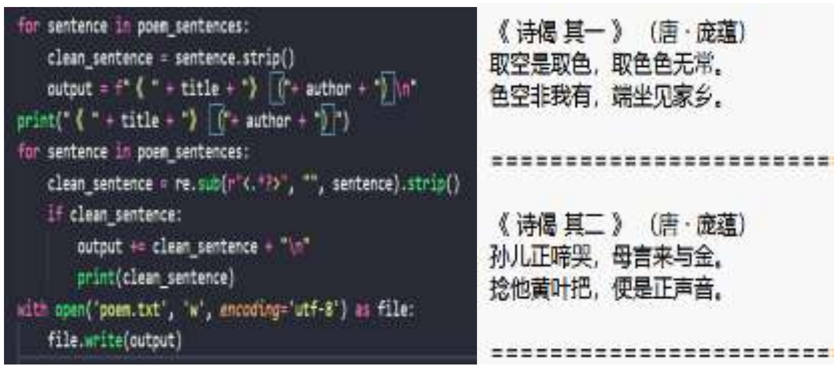
Figure 2. Retrieve and generate answer

Methodology

The first question of generating data sources is where the data comes from. We compared some classical Chinese poetry websites and found that there were many invalid contents and some errors in the content, so we chose school textbooks from different provinces in China as the source of data, and used Spider, Jieba word segmentation, and other technologies to complete our data sources.

Spider: We used the automated program, Spider, for browsing and extracting information over the Internet. Crawlers extract data from web pages and store it by simulating the behavior of human users (Kaur, 2015; Mitchell, 2018), or for further processing. When the crawler starts to work, it first selects a starting web page as its starting point and then gradually passes through and visits other links according to certain rules. The crawler obtains the web content by sending an HTTP request and uses the HTML parser to extract the required data.

This paper uses Requests to send HTTP requests, grab classical Chinese content, and generate TXT-type files. When generating the database, we should keep the information of the title, the author and the dynasty, so we are very careful in designing the capture data code. The Spider code and generated the TXT file are shown in Figure 3.



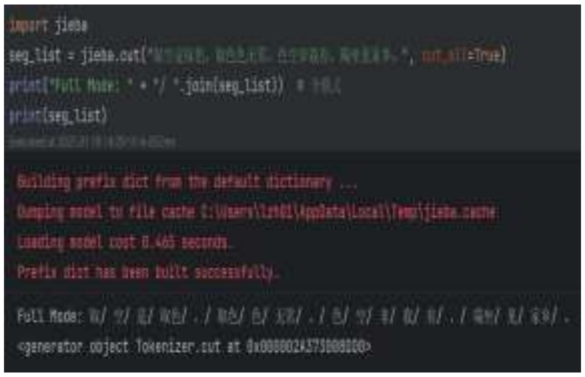
```
for sentence in poem_sentences:
    clean_sentence = sentence.strip()
    output = f"《 " + title + "》 （"+ author + "）\n"
print("《 " + title + "》 （"+ author + "）")
for sentence in poem_sentences:
    clean_sentence = re.sub(r"<.*?>", "", sentence).strip()
    if clean_sentence:
        output += clean_sentence + "\n"
        print(clean_sentence)
with open('poem.txt', 'w', encoding='utf-8') as file:
    file.write(output)
```

```
《诗偈 其一》（唐·庞蕴）
取空是取色，取色色无常。
色空非我有，端坐见家乡。

=======================

《诗偈 其二》（唐·庞蕴）
孙儿正啼哭，母言来与金。
捻他黄叶把，便是正声音。

=======================
```

Figure 3. Extract the classical Chinese content and output a TXT file

Jieba: Jieba is a third-party Chinese word library in Python, known for its simplicity, ease of use, powerful functions, and excellent performance. It is widely used in natural language processing fields, such as text analysis, information retrieval, and machine translation. The core function of Jieba word segmentation is to cut a piece of Chinese text into independent words.

It supports three main modes of word segmentation: Precision mode, which attempts to cut the sentence most precisely, is suitable for scenarios such as text analysis. Full mode: List all possible words in the sentence without considering ambiguity. (Zhao, 2024) Search engine mode: based on the accurate mode, long words are segmented again, improving the recall rate and making it suitable for search engine construction indices. The full mode code is shown in Figure 4.



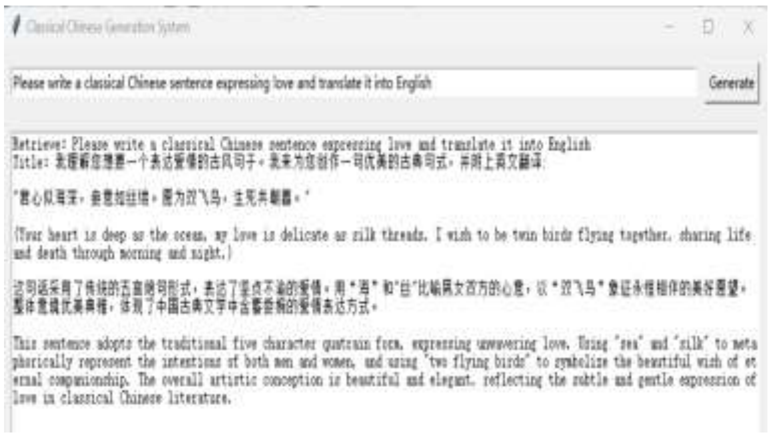
```
import jieba
seg_list = jieba.cut("[illegible]", cut_all=True)
print("Full Mode: " + "/ ".join(seg_list))  # 全模式
print(seg_list)
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\[illegible]\AppData\Local\Temp\jieba.cache
Loading model cost 0.465 seconds.
Prefix dict has been built successfully.

Full Mode: [illegible]
<generator object Tokenizer.cut at 0x000002A375B08DD0>
```

Figure 4. Use Jieba segmentation to divide classical Chinese into nouns and verbs

In addition to the basic word segmentation function, Jieba also provides the following functions: part-of-speech annotation. Jieba can mark the part of each word in the result of word segmentation, such as nouns, verbs, adjectives, etc. This is particularly useful for a more in-depth text analysis.

After word segmentation, sort nouns, verbs, and adjectives according to their frequency, and integrate by words. Other data sources utilize complete classical Chinese works for translation, whereas we choose to translate by word. As a result, the storage space of the generated data sources has increased, but the content is more detailed. It is worth mentioning that when we performed such a quantification, we got a very good response in the subsequent implementation of the RAG method. We chose to use the Qwen LLM to handle Retrieve and Generate. Here is a presentation of the results in Figure 5.



Please write a classical Chinese sentence expressing love and translate it into English

Retrieve: Please write a classical Chinese sentence expressing love and translate it into English
Title: 我理解您想要一个表达爱情的古风句子。我来为您创作一句优美的古典句式，并附上英文翻译。

"君心似海深，妾意如丝缕。愿为双飞鸟，生死共朝暮。"

(Your heart is deep as the ocean, my love is delicate as silk threads. I wish to be twin birds flying together, sharing life and death through morning and night.)

这句话采用了传统的五言绝句形式，表达了坚贞不渝的爱情。用"海"和"丝"比喻男女双方的心意，以"双飞鸟"象征永恒相伴的美好愿望。整体意境优美典雅，体现了中国古典文学中含蓄委婉的爱情表达方式。

This sentence adopts the traditional five character quatrain form, expressing unwavering love. Using "sea" and "silk" to metaphorically represent the intentions of both men and women, and using "two flying birds" to symbolize the beautiful wish of eternal companionship. The overall artistic conception is beautiful and elegant, reflecting the subtle and gentle expression of love in classical Chinese literature.

Figure 5. Generate results for presentation

Results and Discussion

When using large language models to generate classical Chinese, the decision on whether to employ retrieval augmentation generation technology has a significant impact on the quality of the generated results, particularly in two key dimensions: text accuracy and semantic coherence. Due to the limitations of the static knowledge base's coverage, traditional methods often struggle to accurately grasp linguistic phenomena, such as the polysemy of words and the differences in meaning between ancient and modern words in classical Chinese, resulting in problems like historical errors or inappropriate contexts in the generated texts. Given this technical bottleneck, this paper proposes a data source generation method based on multi-semantic extension, which classifies classical Chinese words according to their linguistic features, such as extended meaning, false meaning, and allusion usage, by constructing a multi-dimensional word meaning vector space, to expand the semantic coverage of retrieval resources significantly.

The comparative experimental results shown in Figures 6 and 7 confirm that the proposed method has a substantial improvement in accuracy and consistency compared with the traditional RAG method. These improvements provide a new technical path for researching ancient text generation in the field of computational linguistics and lay a methodological foundation for the intelligent processing of ancient books in digital humanities research. In terms of allusion accuracy, the new method has an accuracy rate of 72%, which is 7 percentage points higher than the traditional method's 65%. In terms of polysemy resolution, the accuracy of the new method is 65%, which is 24 percentage points higher than the 41% of the traditional method. In terms of consistency, the new method can generate 82% of the human mind in classical Chinese, compared to only 51% of the traditional method.

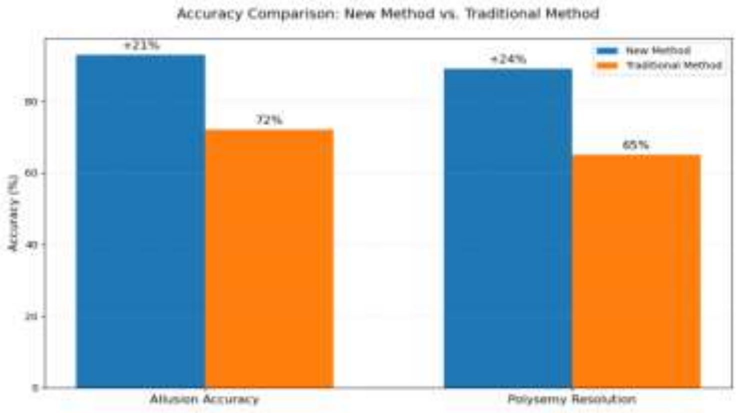


Figure 6. Different methods of accuracy comparison

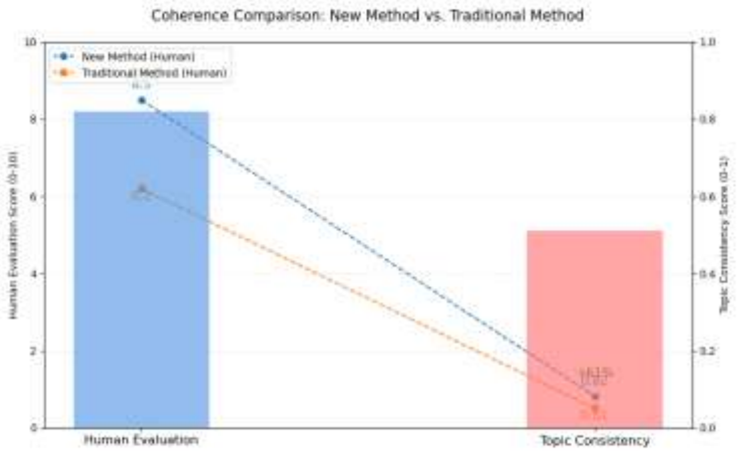


Figure 7. Different methods of coherence comparison

They fine-tune on the existing large model (Ihnaini, 2024), as we said that one of the two ways to solve the limitations is fine-tuning, the advantage of fine-tuning is to get a new large model but also causes a waste of resources. Our approach to using RAG enables existing large models to retrieve new data sources we create, with the same effect, without requiring retraining of the large model.

Conclusions

RAG enables Langchain to retrieve contextually relevant texts from a curated knowledge base (e.g., annotated Confucian classics, dynastic histories), thereby enhancing the language model's ability to generate semantically and historically accurate responses in Classical Chinese. This is critical because Classical Chinese is highly context-dependent and elliptical; the words often carry multiple meanings depending on the era or text. When translating or explaining Classical Chinese texts, RAG retrieves scholarly interpretations, commentaries, or parallel modern Chinese translations. This supports multi-layered understanding and cross-validation of meanings. The application of RAG in Langchain significantly enhances the handling of Classical Chinese texts by providing contextual accuracy, scholarly relevance, and interactive capabilities. It enhances traditional NLP pipelines with retrieval-enhanced comprehension, making it suitable for academic research, cultural preservation, and AI-assisted classical education.

Acknowledgement

Not applicable.

References

Cai, Y. S., Mu, X. Y., Dong, H. C., Guo, Q., & Sun, D. (2023). Optimization of document segmentation method in LangChain framework: Methods and applications. *Computer Science and Application*, *13*(12), 2575–2586. https://doi.org/10.12677/CSA.2023.1312256

Ihnaini, B., Sun, W., Cai, Y., Xu, Z., & Sangi, R. (2024, May). Sentiment analysis of Song Dynasty classical poetry using fine-tuned large language models: A study with LLMs. In *Proceedings of the 7th International Conference on Artificial Intelligence and Big Data* (pp. xx–xx). IEEE. https://doi.org/10.1109/ICAIBD62003.2024.10604440

Jeong, C. (2023). Generative AI service implementation using LLM application architecture: Based on RAG model and LangChain framework. *Journal of Intelligence and Information Systems*, *29*(4), 129–164. https://doi.org/10.13088/jiis.2023.29.4.129

Kaur, S., & Kaur, K. (2015). Web mining and data mining: A comparative approach. *International Journal of Novel Research in Computer Science and Software Engineering*, *2*(1), 36–42. https://www.noveltyjournals.com/issue/IJNRCSSE/Issue-1-January-2015-April-2015

Leong, W. Y. (2025a). Machine learning in evolving art styles: A study of algorithmic creativity. *Engineering Proceedings*, *92*(1), 45. https://doi.org/10.3390/engproc2025092045

Leong, W. Y. (2025b). AI-generated artwork as a modern interpretation of historical paintings. *International Journal of Social Sciences and Artistic Innovations*, *5*(1), Article 0002. https://doi.org/10.35745/ijssai2025v05.01.0002

Leong, W. Y. (2025c, January). AI-powered color restoration of faded historical paintings. Paper presented at the 10th International Conference on Digital Arts, Media and Technology & the 8th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, Nan, Thailand. https://doi.org/10.1109/ECTIDAMTNCON64748.2025.10961986

Liu, M., Xiang, J., Xia, X., & Hu, H. (2022). Contrastive learning between classical and modern Chinese for classical Chinese machine reading comprehension. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(2), Article 1–22. https://doi.org/10.1145/3551637

Mitchell, R. (2018). *Web scraping with Python: Collecting more data from the modern web* (2nd ed.). O'Reilly Media. https://www.oreilly.com/library/view/web-scraping-with/9781491985564/

Pokhrel, S., Ganesan, S., Akther, T., & Karunarathne, L. (2024). Building customized chatbots for document summarization and question answering using large language models with a framework combining OpenAI, LangChain, and Streamlit. *Journal of Information Technology and Digital World*, *6*(1), 70–86. https://doi.org/10.36548/jitdw.2024.1.006

Şakar, T., & Emekci, H. (2025). Maximizing RAG efficiency: A comparative analysis of RAG methods. *Natural Language Processing*, *31*(1), 1–25. https://doi.org/10.1017/nlp.2024.53

Xiang, J., Liu, M., Li, Q., Qiu, C., & Hu, H. (2024). A cross-guidance cross-lingual model on generated parallel corpus for classical Chinese machine reading comprehension. *Information Processing & Management*, *61*(2), 103607. https://doi.org/10.1016/j.ipm.2023.103607

Zhao, X., Huang, J., Zhang, J., & Song, Y. (2024). The comprehensive analysis of the effect of Chinese word segmentation on fuzzy-based classification algorithms for agricultural questions. *International Journal of Fuzzy Systems*, *26*(8), 2726–2749. https://doi.org/10.1007/s40815-024-01724-0