Article

Combining Lexicon Definitions and Retrieval-Augmented Generation of Large Language Model for Automatic Annotation of Ancient Chinese Poetry

Jiabin Li , Tingxin Wei , Weiguang Qu * , Bin Li , Minxuan Feng , Dongbo Wang

Posted Date: 20 May 2025

doi: 10.20944/preprints202505.1554.v1

Keywords: automatic annotation; knowledge base construction; large language model

Article

Combining Lexicon Definitions and Retrieval-Augmented Generation of Large Language Model for Automatic Annotation of Ancient Chinese Poetry

Jiabin Li [1], Tingxin Wei [2], Weiguang Qu [1,3,4,*], Bin Li [1], Minxuan Feng [1] and Dongbo Wang [5]

[1] School of Liberal Arts, Nanjing Normal University, Nanjing 210023, China
[2] School of International Culture and Education, Nanjing Normal University, Nanjing 210023, China
[3] School of Computer and Electronic Information, Nanjing Normal University, Nanjing 210023, China
[4] Zhongbei College, Nanjing Normal University, Nanjing 210023, China
[5] School of Information Management, Nanjing Agricultural University, Nanjing 210095, China
* Correspondence: wgqu_nj@163.com

Abstract: Existing approaches to automatic annotation of classical Chinese poetry often fail to generate precise source citations and depend heavily on manual segmentation, limiting their scalability and accuracy. To address these shortcomings, we propose a novel paradigm that integrates dictionary retrieval with retrieval-augmented large language model enhancements for automatic poetic annotation. Our method leverages the contextual understanding capabilities of large models to dynamically select appropriate lexical senses and employs an automated segmentation technique to minimize reliance on manual splitting. For poetic segments absent from standard dictionaries, the system retrieves pertinent information from a domain-specific knowledge base and generates definitions grounded in this auxiliary data, thereby substantially improving both annotation accuracy and coverage. Experimental results demonstrate that our approach outperforms general-purpose large language models and pre-trained classical Chinese language models on automatic annotation tasks; notably, it achieves a micro-averaged accuracy of 94.33% on key semantic segments. By delivering more precise and comprehensive annotations, this framework advances the computational analysis of classical Chinese poetry and offers significant potential for intelligent teaching applications and digital humanities research.

Keywords: automatic annotation; knowledge base construction; large language model

1. Introduction

Classical Chinese poetry represents a treasure of traditional Chinese culture, embodying profound historical and cultural connotations as well as exquisite artistic techniques. The deep philosophical reflections and aesthetic values embedded in these poems not only constitute a vital subject for literary scholarship but also play a significant role in cultural transmission and aesthetic education in contemporary society. However, because the linguistic features of classical poetry differ markedly from those of modern Chinese—frequently involving allusions, metonymy, semantic shifts, and other compact "chunks" [1]—modern readers often encounter substantial obstacles in comprehension, which in turn seriously hinders the popularization and inheritance of these works.

To address these challenges, it is necessary to provide precise and comprehensive annotations for classical poems. Annotation (笺注) refers to the practice of explicating key components of a poem in order to help readers grasp its content, emotional tone, and cultural significance. Through annotation, readers can more accurately apprehend the poem's central theme while also delving into the meanings of annotated segments and the deeper information they convey. Traditional annotations rely on philological methods—examining character definitions, interpreting historical allusions, and analyzing imagery—to guide readers toward an accurate understanding of poetic meaning. Nevertheless, manual annotation is inherently inefficient and cannot scale to the vast corpus of existing texts; moreover, it is difficult to ensure consistent quality across different annotators. With the rapid advancement of artificial intelligence techniques, research into automated annotation of classical poetry has become an inevitable path for overcoming the bottlenecks of traditional annotation.

Current approaches to automated annotation can be broadly divided into two categories: dictionary-based methods and generative language-model methods. Dictionary-based approaches can supply relatively accurate lexical explanations but cannot dynamically adjust sense selection according to context; they struggle with polysemous terms or context-dependent vocabulary and are unable to define entries absent from the dictionary. Conversely, methods based on generative language models can produce flexible annotations, yet their performance hinges critically on the quality and diversity of their training data, and they suffer from issues such as hallucinations and inconsistent outputs [2]. Furthermore, generative-model approaches often rely on manual identification of text chunks to be annotated, and the granularity of these manually segmented chunks has a pronounced effect on annotation accuracy, rendering large-scale automated annotation infeasible.

In response to these limitations, this study proposes an automated annotation paradigm that combines dictionary definitions with retrieval-augmented generation [3]. Building on the accuracy of dictionary-based explanations, this paradigm integrates the contextual understanding capabilities of large language models to dynamically select the appropriate sense of a term according to its context, thereby addressing the shortcomings of traditional dictionary annotation. Simultaneously, by employing automatic text-chunk segmentation techniques, it eliminates the dependence on manual chunking. For segments not covered in existing dictionaries, the system retrieves relevant information from large-scale domain knowledge bases and generates the optimal annotation. This method not only enhances annotation flexibility and accuracy but also improves the handling of complex contexts, offering a more comprehensive and reliable solution for the automated annotation of classical Chinese poetry.

2. Related Work

2.1. Traditional Annotation Studies

Traditional annotation of classical Chinese poetry employs philological methods—such as etymological analysis of characters, interpretation of historical allusions, and examination of poetic imagery—to explicate and interpret ancient verse. Its primary aim is to assist readers in accurately apprehending the poem's surface meaning through a detailed analysis of lexical choices, syntactic structures, and cultural context, while also uncovering deeper cultural connotations and aesthetic value. Such annotations typically rely on documentary research, historical records, and linguistic expertise, with particular emphasis on tracing and elucidating the origins of allusions, historical events, and cultural symbols.

However, traditional annotation exhibits several significant limitations. First, it depends entirely on manual effort, resulting in low efficiency and an inability to process large corpora. Second, annotation quality varies according to the annotator's individual scholarship and subjective interpretation, owing to the absence of a standardized framework, which leads to inconsistent results. Third, traditional studies tend to focus on canonical texts, paying insufficient attention to lesser-known works, thereby constraining both the breadth and depth of research.

Although a number of educational websites—such as 古诗文网 (Gushiwen.cn) [4] and 古诗词网 (GushiCi.com) [5]—have attempted to digitize traditional annotation resources, their lexical notes largely derive from compilations of expert-authored annotated editions (e.g., *The Annotated Anthology of Li Bai* [6] and *The Complete Annotated Works of Du Fu* [7]). Because these resources are constrained by the scope of human scholarship, they cover only a small fraction of the vocabulary appearing in classical poetry and cannot meet the demand for large-scale automated annotation. These inherent limitations of traditional annotation underscore the necessity for research into efficient, accurate, and scalable methods for automatic annotation.

2.2. Automatic Annotation Research

Automatic annotation of classical Chinese poetry can be classified into two main approaches: dictionary-based annotation and generative language–model–based annotation. Dictionary-based methods—exemplified by the SouYun platform [8]—perform word segmentation on the input text and then retrieve all dictionary senses for each segment. However, this approach has three key limitations. First, it cannot disambiguate a term's context-specific sense; instead, it presents every possible definition for a given lexeme. Second, annotation accuracy depends critically on the quality of the word segmentation: segmentation errors directly degrade the correctness of the retrieved definitions. Third, dictionary-based systems cannot annotate out-of-vocabulary items, since any term absent from the dictionary receives no explanatory entry.

With the advent of large pretrained language models, generative approaches have shown substantial promise. For example, Li Shen et al. (2023) developed "AI Taiyan," a pretrained and fine-tuned Classical Chinese language model capable of sentence boundary disambiguation, allusion recognition, annotation, and prose-vernacular translation [9]. Unlike dictionary-based methods, generative models can select the appropriate sense of a term according to its context, markedly improving annotation flexibility. Yet they too face challenges: (1) they require manual identification of target segments for annotation, and the granularity of these manually extracted chunks significantly influences annotation quality, rendering bulk automation impractical; (2) their outputs are inherently probabilistic, making it difficult to guarantee the precise generation of etymological sources or illustrative examples; and (3) the computational expense of training or fine-tuning such large models is substantial, while issues such as model hallucination (the production of inaccurate or irrelevant content) and unintended shifts in model behavior during fine-tuning remain unresolved.

In summary, although both dictionary-based and generative language–model approaches have advanced the field of automated annotation, each exhibits intrinsic limitations. These shortcomings highlight critical directions for future research aimed at developing methods that are simultaneously more efficient, accurate, and scalable.

2.3. Research on Classical Chinese Poetry and Ci

In recent years, driven by rapid advances in natural language processing (NLP), automated analysis and study of classical Chinese poetry have emerged as a focal point in academia. Researchers have applied deep learning and pretrained language–model techniques to tasks ranging from poetic genre classification and poem generation to sentiment analysis, with the aim of uncovering the underlying linguistic patterns, cultural characteristics, and artistic values of these works.

Hu et al. (2015) conducted automatic thematic classification of Tang‑dynasty poems by extracting features from the poem's text, title, form, and author metadata [10]. Huang et al. (2023) performed a difficulty grading of the *Three Hundred Tang Poems* corpus at the character, word, and sentence levels, employing k-means clustering on a deeply processed text representation to stratify poem difficulty [11]. Liu et al. (2020) created the Ancient Poetry Readability Dataset Plus (APRD+), a manually annotated corpus with six readability tiers. They used features such as character frequency and the number of lexical annotations, and applied machine‑learning algorithms (e.g., random forest) to evaluate reading‑difficulty levels in classical poetry [12].

Yao (2011) developed an automatic allusion‑analysis system by constructing an ontology of historical and literary allusions as a reference knowledge base, enabling the system to identify instances of allusion in poetic text [13]. Tang et al. (2019) reframed allusion recognition in Tang poetry as a relevance–ranking task: for each test sentence, they computed similarity scores against candidate allusion sentences, weighting both single‑character and bi-character overlaps to rank potential allusive references [14].

Yi et al. (2018) introduced a working-memory model for Chinese poetry generation that leverages historical information—such as the poem's theme and key lexical items—to maintain both coherence and thematic consistency across generated lines [15,16].

Despite these advances, most existing studies focus on surface-level textual features—characters, words, sentences, and document-level statistics or semantic embeddings—without fully exploiting deeper, contextually informed semantic representations inherent in classical poetry.

2.4. Large Language Models and Retrieval-Augmented Generation

Recent studies have demonstrated the feasibility of applying general-purpose large language models (LLMs) to downstream information-extraction tasks—such as relation extraction and named-entity recognition—in Chinese text [17–20]. However, because off-the-shelf LLMs lack in-depth expertise in specialized domains, researchers have developed vertical-domain LLMs tailored to legal [21–23] and medical [24–26] applications. Training such models demands vast quantities of high-quality, domain-specific corpora; the collection, curation, and annotation of these resources incur substantial human and computational costs. Moreover, when LLMs generate domain-specific content or citations, their outputs may lack precision or contain errors—a phenomenon commonly referred to as “model hallucination” [27].

To mitigate these limitations, a growing body of work employs retrieval-augmented generation (RAG), a paradigm that supplements generative models with external knowledge retrieved from specialized knowledge bases. In the legal domain, for instance, DISC-LawLLM enhances its responses by dynamically retrieving and integrating relevant statutes or case law passages during generation [28]. In finance, Setty et al. have proposed multiple chunking strategies that index financial documents by segment and keyword, enabling the model to query the appropriate sub-documents in real time to support accurate answer generation [29].

Looking ahead, key research directions include:

(1) Refining retrieval mechanisms—such as improving retrieval precision and reducing latency.

(2) Constructing and maintaining high-fidelity domain knowledge bases with rigorous provenance and version control.

(3) Advancing LLM architectures to better integrate and reason over deep semantic and factual knowledge. Progress in these areas will not only bolster the factual accuracy and relevance of generated content but also enable LLMs to tackle high-precision, time-sensitive tasks—thereby laying a robust foundation for intelligent, knowledge-driven services.

3. Automatic Annotation Method

Current expert-annotated corpora for classical Chinese poetry are limited in size and lack a unified annotation standard. Moreover, existing annotation datasets are both insufficient in volume and unevenly distributed—focusing primarily on canonical poets and works while neglecting marginal texts. Training an automatic annotation model directly on such imbalanced data would inevitably impair both its accuracy and its ability to generalize. To address these issues, we propose a method that combines dictionary-based knowledge with a large language model (LLM), leveraging the LLM's strengths in contextual text processing and selecting the most appropriate sense from a reference dictionary according to poetic context. The overall research framework is illustrated in Figure 1.

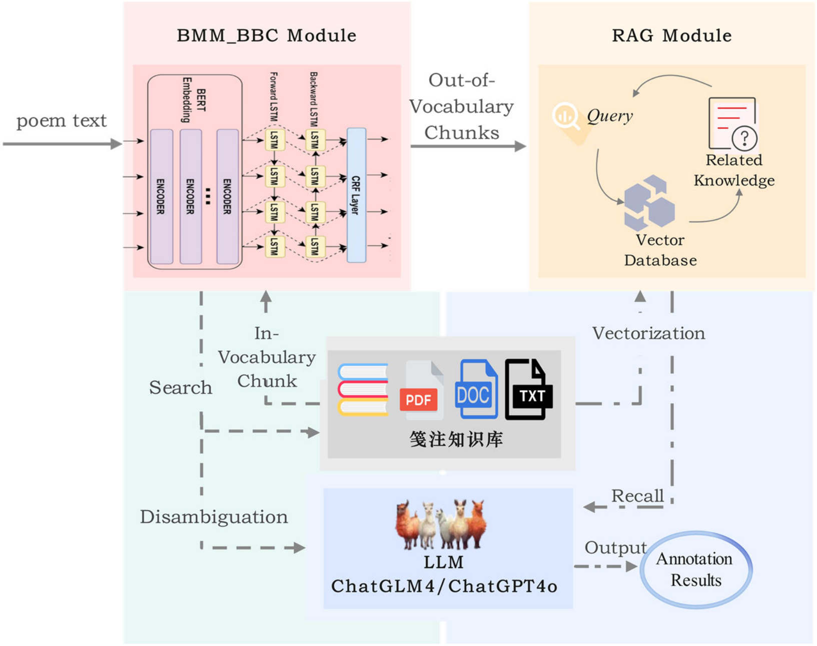


Figure 1. Automatic Annotation Research Framework.

In our workflow, manual chunk selection—without genuine understanding of each chunk's meaning—fails to guarantee correct segmentation and cannot scale to large-volume annotation. Therefore, we first apply the BMM_BBC Chunking Module to segment the input poem automatically. For each resulting chunk, we query a curated lexical resource: if dictionary definitions exist, we pass the chunk and its candidate senses into the LLM for disambiguation; if a chunk is not found in the dictionary (i.e., out-of-vocabulary), we invoke our retrieval-augmented generation (RAG) module to retrieve relevant domain knowledge and generate an optimal annotation.

3.1. BMM_BBC Chunking Module

Accurate segmentation of semantic chunks is critical for subsequent sense retrieval and annotation generation and is a prerequisite for scalable, bulk automatic annotation. Existing chunking methods in the classical-poetry domain face two main challenges: (1) Deep-learning-based approaches (e.g., the BERT-BiLSTM-CRF "BBC" model) capture rich contextual information but are hampered by the scarcity of large, domain-specific annotated corpora; (2) Dictionary-based methods (e.g., bidirectional maximum matching, "BMM") exploit prior knowledge from poetry dictionaries but cannot handle out-of-vocabulary terms.

To harness the advantages of both, we introduce BMM_BBC, a hybrid segmentation method. First, the BMM algorithm produces an initial segmentation, which serves as prior knowledge. We then apply a set of linguistically motivated rules to filter and weight the BMM output, integrating these weights into the BBC model's decoding process. In doing so, the BBC model dynamically adjusts its segmentation decisions under the influence of dictionary priors. This hybrid strategy preserves the BBC model's contextual sensitivity while enhancing its ability to recognize and correctly segment previously unseen terms—achieving higher-quality chunking without reliance on extensive, manually annotated corpora.

In Figure 2, The example sentence is drawn from the Tang- dynasty poem *Princess Anle's Relocation to Her New Residence*, composed to commemorate Princess Anle's move into her newly appointed dwelling. the italicized and underlined segments represent literary allusions: 铺钱埒 ("coin-paved enclosure," a metaphor for extravagant living) and 弄玉台 ("Phoenix Terrace," originally the site of celestial ascension, here metonymically denoting the princess's residence), both of which are registered entries in the allusion dictionary. The text shown in red indicates positions where the predicted labels are incorrect.

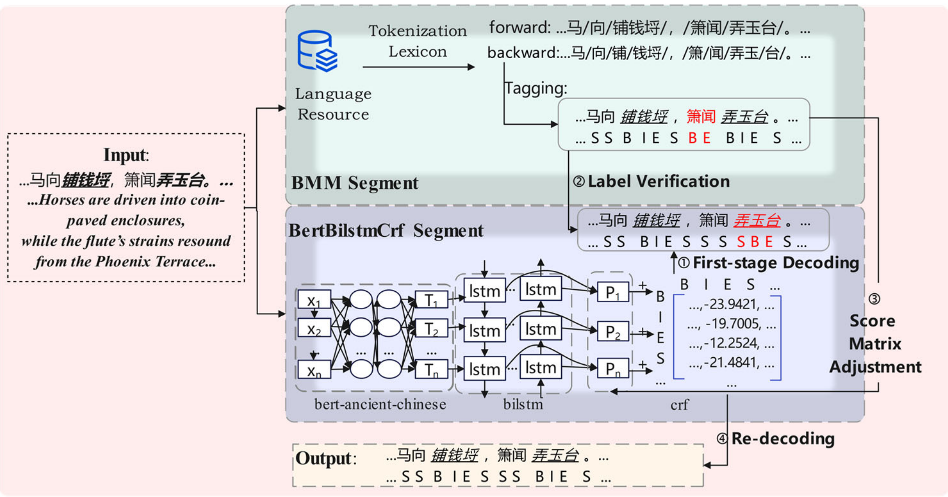

Tokenization
Lexicon
forward: ...马/向/铺钱埒/，/箫闻/弄玉台/。...
backward:...马/向/铺/钱埒/，/箫/闻/弄玉/台/。...
Language
Resource
Tagging:
...马向 铺钱埒，箫闻 弄玉台。...
... S S B I E S B E B I E S ...
Input:
...马向铺钱埒，箫闻弄玉台。...
...Horses are driven into coin-paved enclosures,
while the flute's strains resound
from the Phoenix Terrace...
BMM Segment
② Label Verification
BertBilstmCrf Segment
...马向 铺钱埒，箫闻 弄玉台。...
... S S B I E S S S S B E S ...
① First-stage Decoding
B I E S ...
...,-23.9421, ...
..., -19.7005, ...
...,-12.2524, ...
...,-21.4841, ...
③
Score
Matrix
Adjustment
x1
x2
xn
T1
T2
Tn
lstm
P1
P2
Pn
bert-ancient-chinese
bilstm
crf
④ Re-decoding
Output:
...马向 铺钱埒，箫闻 弄玉台。...
... S S B I E S S S B I E S ...

Figure 2. BMM_BBC Chunking Module.

3.1.1. Training the BBC Model

The BBC model comprises three primary components: a BERT encoding layer, a bidirectional LSTM (BiLSTM) layer, and a conditional random field (CRF) layer. BERT (*Bidirectional Encoder Representations from Transformers*) is a pretrained language model introduced by Google in 2018. It is built on the Transformer architecture and employs a bidirectional encoder, thereby capturing contextual information from both the left and right of each token [30].

For sequence labeling, we use a CRF to predict the most probable label sequence $y=\{y_1, y_2, ..., y_t\}$ given an input X of length t. The score of the entire label sequence is defined as:

$$s(X,y)=\sum_{i=0}^{t} A_{y_i,y_{i+1}} + \sum_{i=1}^{t} p_{i,y_i} \qquad (1)$$

where y_0 and y_{i+1} denote the designated start and end labels, respectively; p_{i,y_i} is the emission score for assigning label y_i at position *I*; and *A* is the transition score matrix, with $A_{j,k}$ representing the score of transitioning from label *j* to label *k*. We convert these scores into a conditional probability via the softmax over all possible label sequences:

$$P(y\,|\,X)=\frac{\exp(s(X,y))}{\sum_{y'\in y}\exp(s(X,y'))} \qquad (2)$$

Regarding the training corpus, both classical Chinese poetry and prose originate from the same historical periods and conform to the grammatical conventions of Old Chinese. Given the lack of large-scale, manually segmented corpora for classical poetry and the absence of contemporaneous prose segmentation data, we construct a hybrid segmentation corpus by combining the LDC *Zuozhuan* word-segmentation dataset [31] with segmentation data from the *Three Hundred Tang Poems*. We then conduct full-parameter training of the BBC model on this mixed corpus, yielding the optimized model weights, denoted γ.

3.1.2. Semantic Chunking with Integrated Dictionary Information

To combine the BMM and BBC segmentation outputs, we adjust the model's initial emission score matrix P based on the label sequence produced by BMM, thereby biasing the decoder toward segments consistent with the dictionary.

Given an input sentence X, we apply forward and backward maximum-matching (BMM) to obtain a segmentation

$$W=\{w_1, w_2, ..., w_k\} \qquad (3)$$

where each w_k is a contiguous token from the dictionary. Then, the segmented sequence W is cconverted into a BMM label sequence

$$L_{BMM} = \{l_1, l_2, ..., l_t\} \qquad (4)$$

where each label is drawn from the set {B,M,E,S}, denoting Begin, Middle, End, or Single-token segments.

Using the unmodified emission score matrix P produced by the BBC model, we apply the Viterbi algorithm to compute an initial predicted label sequence.

$$L_{BBC} = Viterbi(s(X, y)) \qquad (5)$$

We then adjust each emission score p_{i,y_i} in P according to the corresponding BMM label. The adjusted score is given by

$$\hat{p}_{i,y_i} = p_{i,y_i} + \delta(1_i, y_i) \qquad (6)$$

where the increment δ is determined by the rule set in Formula 7. This adjustment encourages the CRF decoder to favor segments that align with dictionary-based priors, while still allowing the BBC model's contextual understanding to resolve ambiguous cases.

$$\delta(1_i, y_i) == \begin{cases} \lambda_1, & \mathrm{r1} \\ \lambda_2, & \mathrm{r2} \\ 0, & \mathrm{r3} \end{cases} \qquad (7)$$

Rule r_1 (Forced Adjustment). If a BMM-derived token w_k has length ≥3, which is a strong indicator that it represents an allusion or proper-name chunk—and the BMM label l_k^{BMM} disagrees with the BBC label l_k^{BBC} , then for every position i within that token, we apply a nonzero increment δ to the emission score. This enforces the dictionary-based segmentation in cases where long tokens likely correspond to fixed cultural or named entities.

Rule r_2 (General Adjustment). If a BMM token w_k of any length corresponds to an entry present in the dictionary (regardless of length) and there is any discrepancy with the BBC segmentation, we add a smaller positive increment δ uniformly to all positions i of that token. This encourages the model to favor known, in-dictionary segments during decoding.

Rule r_3 (No Adjustment). In all other cases—i.e., when a BMM segment is neither long (⩾ 3) nor a recognized dictionary entry—we set δ=0, leaving the original emission scores unmodified.

By substituting the adjusted emission scores into the CRF sequence-scoring function, we obtain the updated sequence score:

$$\hat{s}(X, y) = \sum_{i=0}^{t} A_{y_i, y_{i+1}} + \sum_{i=1}^{t} (p_{i,y_i} + \delta(l_i, y_i)) \ (8)$$

Subsequently, we apply the Viterbi algorithm to this revised scoring function to derive the adjusted predicted label sequence:

$$L_{BMM\_BBC} = Viterbi(\hat{s}(X, y)) \qquad (9)$$

Through extensive experimentation, we empirically set the hyperparameter λ_1 in rule r_1 to 50 and λ_2 in rule r_2 to 20.

3.2. Retrieval-Augmented Generation (RAG) Module

To address the annotation of out-of-vocabulary chunks, we introduce a Retrieval-Augmented Generation (RAG) module. The core idea of RAG is to enhance the generative model's output by retrieving relevant information from external knowledge bases—such as specialized dictionaries and critical commentaries—thereby producing accurate, contextually appropriate annotations for previously unseen segments.

The integration of the RAG module is motivated by two considerations. First, in classical Chinese poetry, certain chunks may not appear in existing lexicons or language resources, rendering traditional dictionary-based methods unable to supply definitions. By querying external repositories, RAG effectively bridges this gap. Second, although large language models excel at processing lengthy texts and complex contexts, their outputs can lack precise grounding in domain-specific knowledge—such as literary allusions and historical backgrounds. By incorporating retrieved facts and examples into the generation process, the RAG module substantially improves both the accuracy and reliability of the resulting annotations.

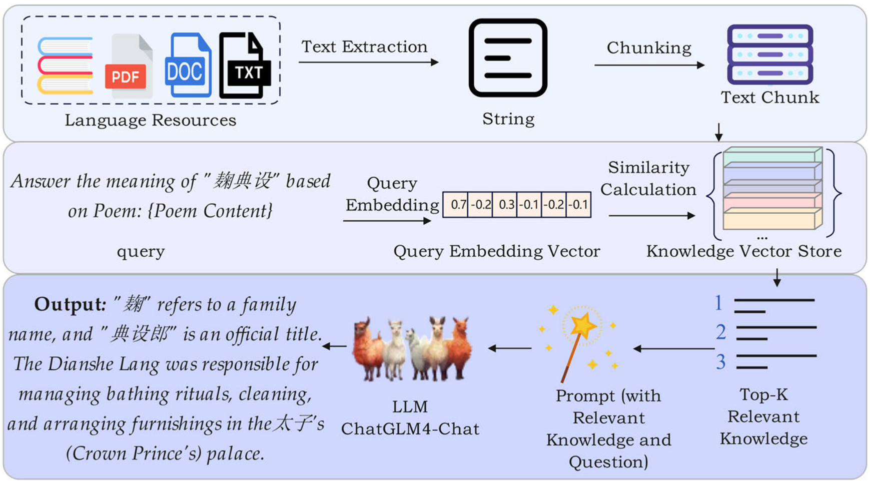

Language Resources
PDF
DOC
TXT
Text Extraction
String
Chunking
Text Chunk
Answer the meaning of "麹典设" based on Poem: {Poem Content}
query
Query Embedding
0.7 -0.2 0.3 -0.1 -0.2 -0.1
Query Embedding Vector
Similarity Calculation
...
Knowledge Vector Store
Output: "麹" refers to a family name, and "典设郎" is an official title. The Dianshe Lang was responsible for managing bathing rituals, cleaning, and arranging furnishings in the 太子's (Crown Prince's) palace.
LLM
ChatGLM4-Chat
Prompt (with Relevant Knowledge and Question)
1
2
3
Top-K Relevant Knowledge

Figure 3. RAG Retrieval-Augmented Generation Module.

In this paper, the construction workflow of the Retrieval-Augmented Generation (RAG) module is formalized by the following equations. Given a set of language resources[1] $T=\{T_1,T_2,\dots,T_n\}$, where each T_i denotes a text document, each document is segmented into fixed-length chunks as follows:

$$T_i = \{C_{i,1}, C_{i,2}, ..., C_{i,m}\} \qquad (10)$$

Let $C=\{C_1,C_2,\dots,C_k\}$ represent the collection of all chunks. Each chunk $C_j \in C$ is encoded into a vector representation v_j using a text encoder f_{text}, resulting in the set of chunk embeddings:

$$v_j = f_{text}(C_j), v_j \in \mathbb{R}^d \qquad (11)$$

$$v_{text} = \{v_1, v_2, ..., v_k\} \qquad (12)$$

Next, the same text encoder f_{text} is used to encode the input query into a vector representation v_q . The similarity score s_j between v_q and each chunk vector v_q is then computed:

$$v_q = f_{text}(query), v_q \in \mathbb{R}^d \qquad (13)$$

Based on the similarity scores, the *top-k* most similar text vectors are selected in descending order of similarity.

$$v_{top-k} = \{v_{j1}, v_{j2}, ..., v_{jk}\} \qquad (14)$$

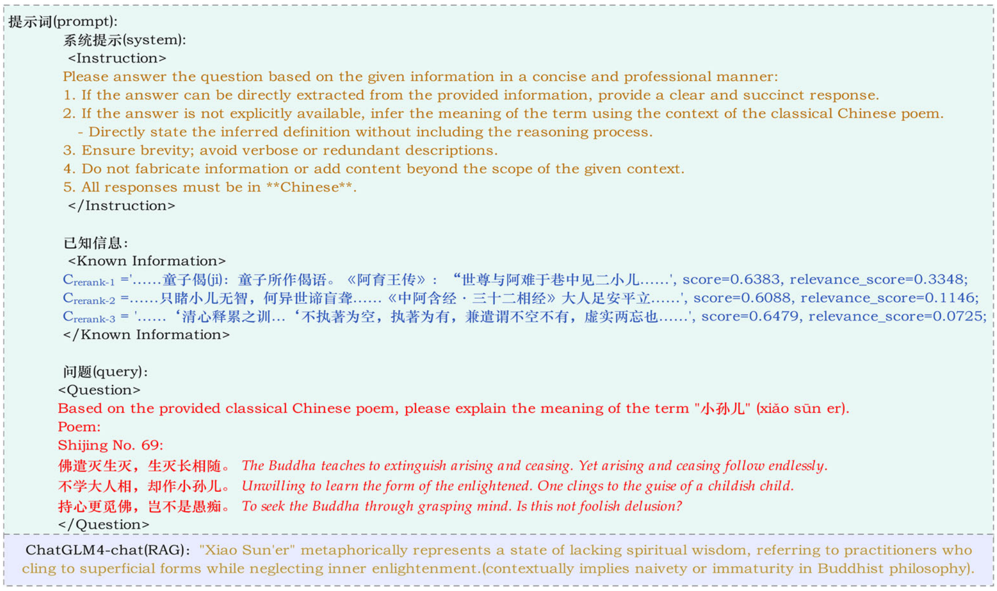
The corresponding chunks of the top-k vectors form the candidate context set $C_{top\text{-}k}=\{C_{j1},C_{j2},...,C_{jk}\}$. These candidates are further re-ranked using a re-ranking model f_{rerank}, which assesses the relevance between each chunk and the query, producing a refined context set C_{rerank}:

$$C_{rerank} = f_{rerank}(query, C_{top-k}) \qquad (15)$$

The re-ranked context set C_{rerank} is then combined with the original query to construct a prompt, which is subsequently passed to a large language model to generate the final output. Figure 4 illustrates an example prompt generated by the RAG module in this study. The "Known Information" section corresponds to the re-ranked context set C_{rerank}, while the initial retrieval scores (i.e., pre-re-ranking similarity scores) are denoted as score. After applying the re-ranking model f_{rerank}, each candidate document receives a new relevance_score based on its semantic alignment with the query, which leads to a refined ranking. For example, although the chunk $C_{rerank\text{-}3}$ does not contain information relevant to "小儿相", it initially received the highest semantic similarity score (0.6479) in the preliminary retrieval stage. However, after re-ranking, its relevance_score drops to 0.0725, demonstrating a more fine-grained and semantically aware relevance evaluation.

Finally, the re-ranked documents are combined with the query to form a prompt for the large language model, which generates the final answer.

doi:10.20944/preprints202505.1554.v1



```
提示词(prompt):
        系统提示(system):
         <Instruction>
        Please answer the question based on the given information in a concise and professional manner:
        1. If the answer can be directly extracted from the provided information, provide a clear and succinct response.
        2. If the answer is not explicitly available, infer the meaning of the term using the context of the classical Chinese poem.
           - Directly state the inferred definition without including the reasoning process.
        3. Ensure brevity; avoid verbose or redundant descriptions.
        4. Do not fabricate information or add content beyond the scope of the given context.
        5. All responses must be in **Chinese**.
         </Instruction>

        已知信息:
         <Known Information>
        $C_{rerank-1}$ ='......童子偈(ji): 童子所作偈语。《阿育王传》: “世尊与阿难于巷中见二小儿......', score=0.6383, relevance_score=0.3348;
        $C_{rerank-2}$ =......只睹小儿无智，何异世谛盲聋......《中阿含经·三十二相经》大人足安平立......', score=0.6088, relevance_score=0.1146;
        $C_{rerank-3}$ = '......‘清心释累之训...‘不执著为空，执著为有，兼遣谓不空不有，虚实两忘也......', score=0.6479, relevance_score=0.0725;
        </Known Information>

        问题(query):
       <Question>
       Based on the provided classical Chinese poem, please explain the meaning of the term "小孙儿" (xiǎo sūn er).
       Poem:
       Shijing No. 69:
       佛遣灭生灭，生灭长相随。 The Buddha teaches to extinguish arising and ceasing. Yet arising and ceasing follow endlessly.
       不学大人相，却作小孙儿。 Unwilling to learn the form of the enlightened. One clings to the guise of a childish child.
       持心更觅佛，岂不是愚痴。 To seek the Buddha through grasping mind. Is this not foolish delusion?
       </Question>
   ChatGLM4-chat(RAG): "Xiao Sun'er" metaphorically represents a state of lacking spiritual wisdom, referring to practitioners who
  cling to superficial forms while neglecting inner enlightenment.(contextually implies naivety or immaturity in Buddhist philosophy).
```

Figure 4. RAG Retrieval-Augmented Generation Module: Prompt Example.

In this study, the RAG module is configured as described in Table 1. Extracted language resources are segmented into fixed-length chunks of 250 tokens. We employ the “bge-large-zh-v1.5” embedding model [32] to encode each chunk for semantic retrieval, and the “bge-reranker-base” model [32] to re-rank the top retrieved candidates, thereby improving recall quality. Finally, we use the “glm-4-9b-chat” model [33] as the generative backbone.

Table 1. Configuration of the RAG Module.

Property	Value
EMBEDDING_MODEL	bge-large-zh-v1.5
RERANKER_MODEL	bge-reranker-base
LLM_MODEL	glm-4-9b-chat
CHUNK_SIZE	250
TOP_K	3
SCORE_THRESHOLD	1
TEMPERATURE	0.7
MAX_TOKENS	2048

4. Results

To comprehensively evaluate the proposed framework— which integrates dictionary knowledge with retrieval-augmented generation for automatic annotation of classical Chinese poetry— we conducted experiments on two fronts: automatic chunk segmentation and automatic chunk annotation.

4.1. Chunk Segmentation Experiment and Analysis

Accurate segmentation of semantic chunks is decisive for subsequent sense retrieval. However, not every multi-character unit must be treated as an inseparable whole, nor is there a universally applicable “gold standard” for all cases. For example, the word “秋日” (“autumn day”) can be segmented into “秋/日” and still yield correct annotations for each character, with its overall meaning preserved. In contrast, for chunks whose holistic meaning differs from the sum of their parts—such as semantic shifts, literary allusions, or proper names—mis-segmentation can lead to grossly incorrect definitions or retrieval failures. Examples include: 泬寥 (“clear and spacious appearance”), 长卿病渴 (a metaphor for a disillusioned scholar living in poverty and illness), 紫府 (in Daoist lore, the dwelling place of immortals). Accordingly, our evaluation specifically targets these critical, non-compositional chunks, measuring the segmentation accuracy for units whose correct boundaries are essential to preserving intended poetic meaning.

4.1.1. Experimental Setup and Evaluation Metrics

We randomly selected 100 poems from the *Quan Tang Shi* corpus and manually identified all semantic chunks for which holistic meaning ≠ compositional meaning, recording each chunk's index position within its line. To assess the segmentation method's performance across varying chunk lengths, we categorized chunks into three types by character count: two-character, three-character, and four-character or longer. For each category $c \in \{2,3,\geq 4\}$, we compute the category-specific accuracy Acc_c as:

$$Acc_c = \frac{1}{N}\sum_{i=1}^{N}\frac{true_c^{(i)}}{count_c^{(i)}} \qquad (16)$$

where N is the number of test samples, $true_c^{(i)}$ denotes the number of correctly segmented chunks of length c in sample i, and $count_c^{(i)}$ is the total number of chunks of length c in sample i.

Next, we calculate a weight w_c for each chunk length according to its proportion of occurrences in the entire test set. We then derive a single, composite metric—Weighted_Accuracy—by combining the per-category accuracies with these weights:

$$Weighted\_Accuracy = \sum_{c=2}^{K} w_c * Acc_c \qquad (17)$$

This weighted measure balances the high frequency of short chunks against the critical importance of accurately segmenting longer, non-compositional units.

4.1.2. Chunk Segmentation Experiments and Analysis

Table 2 reports the segmentation accuracies (Acc) of three methods—BMM, BBC, and BMM_BBC on chunks of length 2, length 3, and length ≥ 4, as well as their overall Weighted_Accuracy.

Table 2. Chunk Segmentation Experiments.

Model	Length = 2 (Acc)	Length = 3 (Acc)	Length ≥ 4 (Acc)	Weighted_Acc
BMM	78.47%	33.74%	43.48%	72.92%
BBC	78.27%	55.19%	53.25%	75.07%
BMM_BBC	92.88%	71.72%	76.46%	90.25%

BMM performs reasonably on two-character chunks (Acc = 78.47%) but struggles with longer units—only 33.74% on three-character and 43.48% on four-character-or-longer chunks—resulting in a modest weighted accuracy of 72.92%. This indicates its weakness in handling longer or more complex segments. BBC substantially improves accuracy on three-character (55.19%) and ≥ four-character (53.25%) chunks by leveraging contextual information, raising its weighted accuracy to 75.07%. However, due to the limited size of domain-specific corpora, it matches BMM on two-character chunks (78.27%) and still exhibits errors in both over- and under-segmentation.

The hybrid model (BMM_BBC) achieves superior performance across all chunk lengths: 92.88% on two-character, 71.72% on three-character, and 76.46% on ≥ four-character chunks, with a combined weighted accuracy of 90.25%. These results demonstrate that BMM_BBC effectively integrates dictionary priors and deep model contextual understanding, substantially enhancing segmentation quality even in the absence of large-scale annotated data.

4.2. Chunk Annotation Experiments and Analysis

To conduct a detailed evaluation of our annotation framework, we assessed its performance on five types of semantic chunks in classical Chinese poetry—allusion, imagery, metonymy, semantic shift, and proper name—using a test corpus of 100 poems randomly sampled from the *Quan Tang Shi*.

4.2.1. Experimental Setup and Evaluation Metrics

We select ChatGPT-4o as our baseline model, owing to its state-of-the-art contextual understanding and generative capabilities. For each of the 100 sampled poems, we manually annotate all instances of the five target chunk types to serve as ground-truth. The annotation performance for each chunk type is measured by the ratio of correctly annotated chunks to the total number of that chunk type. We evaluate performance under two correctness criteria—Strict Correctness and Lenient Correctness—with type-specific definitions as shown in Table 3.

Table 3. Criteria for Strict and Lenient Correctness.

Chunk Type	Strict Correctness	Lenient Correctness
Allusion	Definition and Extended Meaning	Source of the Allusion
Metonymy	Contextual meaning	Attributes of the Referent
Imagery	Emotional or object interpretation	Object only
Semantic Shift	Contextual meaning	Partial contextual meaning
Proper Name	No Erroneous Attributes	Some Erroneous Attributes

Following the evaluation method of Li Shen et al. [12], we define the lenient accuracy for each chunk type as the sum of strictly correct and leniently correct annotations divided by the total number of chunks of that type. To obtain an overall performance metric, we compute the micro-averaged accuracy by summing the number of strictly correct annotations across all types and dividing by the total number of annotated chunks in the corpus.

4.2.2. Comparative Analysis of Chunk Annotation

To compare the annotation performance of a general-purpose large language model (ChatGPT-4o, serving as the baseline), the Taiyan Classical Chinese model (Taiyan 2.0), and our proposed framework, we evaluated each model's accuracy on five semantic chunk types—allusion, metonymy, imagery, semantic shift, and proper name—under both strict and lenient correctness criteria. Table 4 summarizes the results for each model, where "Ours – RAG" denotes our framework with the Retrieval-Augmented Generation (RAG) module removed.

Among the five chunk types, allusions demand the highest level of background knowledge and contextual understanding. Both ChatGPT-4o and Taiyan 2.0 achieve their lowest strict and lenient accuracies on allusion chunks, performing significantly worse than "Ours – RAG" (89.41%) and our full model (90.59%). This gap indicates that the improved performance on allusion annotation primarily stems from the inclusion of entries in the specialized allusion dictionary. In contrast, general-purpose and purely generative text models struggle to capture the rich cultural background and deep connotations of literary allusions; their explanations often reduce to brief citations of source texts and fail to convey interpretive or extended meanings. Consequently, ChatGPT-4o and Taiyan 2.0 exhibit substantial discrepancies between their strict and lenient accuracy scores on allusion chunks.

Table 4. Results of Chunk Annotation.

Chunk	Metric	ChatGPT-4o (%)	Taiyan 2.0 (%)	Ours – RAG (%)	Ours (%)
Allusion	Strict	32.14	32.14	89.41	90.59
	Lenient	65.48	73.81	91.76	94.12
Metonymy	Strict	80.12	85.03	78.47	91.47
	Lenient	89.03	95.21	82.19	95.43

Imagery	Strict	67.74	69.61	74.74	83.16
	Lenient	80.65	80.39	80.00	90.53
Proper Name	Strict	51.15	54.26	66.20	86.57
	Lenient	76.04	75.34	68.52	95.83
Semantic Shift	Strict	75.15	75.15	82.25	90.27
	Lenient	85.63	88.96	84.62	92.63
Micro-Average	Strict	69.28	72.01	77.81	89.72
	Lenient	84.06	86.73	80.93	94.33

Our full model achieves a strict accuracy of 83.16% on imagery chunks, substantially outperforming ChatGPT-4o (67.74%) and Taiyan 2.0 (69.61%). In lenient evaluation, it further improves to 90.53%, compared to 80.65% and 80.39%, respectively. Error analysis reveals that misinterpretations often stem from inadequate context modeling—for example, in the line "谁家稚女著罗裳，红粉青眉娇暮妆," ("*Whose maiden fair in silken dress, With rosy cheeks and brows of jet, Adorns her dusk with tender grace?* ") both "眉" ("*eyebrow*") and "山" ("*mountain*") can be used metaphorically in classical poetry, but annotating "青眉"("*indigo-painted eyebrows*") as "*mountain*" here is clearly incorrect.

Although ChatGPT-4o (80.12% strict) and Taiyan 2.0 (85.03% strict) perform reasonably well under the strict criterion, our model attains 91.47%, leading by a significant margin. The lenient accuracy likewise favors our approach at 95.43%, above ChatGPT-4o's 89.03% and Taiyan 2.0's 95.21%. Notably, the inclusion of the RAG module yields improvements of +13.00% (strict) and +13.24% (lenient) over "Ours – RAG."

Our model excels on proper-name chunks, achieving 86.57% strict accuracy and 95.83% lenient accuracy—vastly higher than ChatGPT-4o (51.15% strict, 76.04% lenient) and Taiyan 2.0 (54.26% strict, 75.34% lenient). This gain is primarily attributable to the specialized proper-name dictionary: the RAG module alone contributes +20.37% (strict) and +27.31% (lenient), the largest single-type improvement observed.

On semantic-shift chunks, our model attains 90.27% strict and 92.63% lenient accuracy, significantly surpassing ChatGPT-4o (75.15% strict, 85.63% lenient) and Taiyan 2.0 (75.15% strict, 88.96% lenient).

Collectively, these results demonstrate that our dictionary-augmented, retrieval-enhanced annotation paradigm consistently improves accuracy across all semantic chunk types—particularly on chunks requiring deep cultural or historical knowledge, such as proper names and allusions. Our approach outperforms both the leading general-purpose baseline and a state-of-the-art classical-Chinese model, underscoring the value of RAG in culturally rich domains and offering a robust methodology for future text generation and interpretability research in Old Chinese and other knowledge-intensive fields.

4.2.3. Comparative Analysis of Source Citations

Table 5 compares each model's ability to identify and output authoritative source citations for annotated chunks. Our full model vastly outperforms both the baseline ChatGPT-4o and the Taiyan 2.0 model in both the total number of cited sources and the number of valid citations. Specifically, out of 1,217 annotated chunks, our model provides source information for 652 instances, of which 613 are verified as valid. In contrast, ChatGPT-4o cites only 18 sources (12 valid), and Taiyan 2.0 cites 23 sources (21 valid).

Table 5. Results of Chunk Source Citations Annotation.

Model	Cited Sources	Valid Citations	Total Annotated Chunks
ChatGPT-4o (baseline)	18	12	1,217
Taiyan 2.0	23	21	1,217

Ours	652	613	1,217

This significant gap arises from our annotation strategy: for in-dictionary chunks, the model not only selects the optimal sense based on context but also automatically retrieves and appends corresponding classical citations and illustrative examples.

During evaluation, we observed that ChatGPT-4o and Taiyan 2.0 generate source citations only for allusion chunks. For instance, in the annotation of "大树" (referring to the historical figure Feng Yi) from the line "大树思冯异," ChatGPT-4o supplies only a biographical summary of Feng Yi and fails to provide the original classical text reference; its explanation of why "大树" refers to Feng Yi is also incorrect. Taiyan 2.0 cites the *Hou Han Shu ("Biography of Feng Yi")* as the source but omits any explanatory gloss, resulting in incomplete annotation. In contrast, our model delivers both accurate citations and coherent, context-grounded definitions.

Our model not only supplies detailed glosses—such as "refers to the Eastern Han general Feng Yi (courtesy name Gongsun), who was appointed 'General Who Assists the Army'"—but also precisely cites the original passage from *Hou Han Shu, Vol. 17 ("Biography of Feng Yi")*, thereby guaranteeing both the completeness and accuracy of its source information. Moreover, the model can generate citations for chunk types beyond literary allusions. For reasons of brevity, Table 6 presents the source annotation for the imagery chunk "浮图" as an imagery example. For lexemes or phrases of significant citation value in classical poetry or historical literature, this capability more fully preserves their cultural and historical context, endowing the automatic annotations with both precision and traceability.

Table 6. Source Citation Examples.

Model	Chunk	Definition	Source
ChatGPT-4o	***大树***思冯异 *The towering tree recalls General Feng Yi.*	In this line, the image of the "big tree" evokes the outstanding talent and notable deeds of Feng Yi.	Cited from the "*Biography of Feng Yi.*" Feng Yi was a renowned general of the early Eastern Han dynasty, celebrated for governing his troops with civil virtue and repeatedly distinguishing himself in battle.
Taiyan 2.0	***大树***思冯异 *The towering tree recalls General Feng Yi.*	None	*Hou Han Shu (Book of Later Han), "Biography of Feng Yi"*: "…每所止舍，诸将并坐论功，异常独屏树下，军中号曰'大树将军'。"
Ours (full)	***大树***思冯异 *The towering tree recalls General Feng Yi.*	Refers to the Eastern Han general Feng Yi (courtesy name Gongsun), who was appointed "General Who Assists the Army." Whenever other commanders convened to discuss merits, he alone would withdraw to stand beneath a tree, earning him the moniker "General of the Big Tree." In later usage, it became an idiom praising exceptional military leaders.	*Hou Han Shu, Vol. 17, "Biography of Feng Yi":* "异为人谦退不伐，行与诸将相逢，辄引车避道。进止皆有表识，军中号为整齐。每所止舍，诸将并坐论功，异常独屏树下，军中号曰'大树将军'。"
Ours (full)	慈恩寺***浮图***应制 *Cien Pagoda: A Poem Composed by Imperial Command*	"浮图" (also written "浮屠") is a Buddhist term, a phonetic transliteration of the Sanskrit "Buddha," referring to a stupa.	*Song dynasty, Record of the Five Hundred Arhats of Jiancheng Chan Yuan:* "…且造铁浮屠十有三级，高百二十尺。"

4.2.4. Inference of Chunk Referents and Annotation Example Analysis

In classical Chinese poetry, understanding chunks such as proper nouns and metonymic expressions often depends heavily on contextual cues and the historical background of the composition. Accurately identifying the referent or context-specific meaning of such chunks is essential for correct semantic interpretation. This section presents a detailed analysis of how the annotation framework proposed in this study performs in inferring referents of such chunks, using a concrete example.

Existing models often struggle with referential chunks, leading to inaccurate glosses, unclear referents, or inconsistent interpretations. In the table 7, red text indicates incorrectly generated content, dark orange highlights portions that remain undefined, and blue text represents correct interpretations. For Example 1, the Taiyan 2.0 model mistakenly interprets “王使君” (Wáng Shǐjūn) as a poet surnamed Li, which is clearly erroneous. Although ChatGPT-4o is able to infer from context that “王使君” refers to a local official stationed in Dongyang, it fails to propose a concrete referent.

To address these challenges, the annotation framework introduced in this paper combines dictionary-based retrieval with large language model reasoning. When encountering referential chunks, the system automatically queries a curated knowledge base to retrieve relevant candidate identities. In the case of “王使君,” the system identifies “王俺” (Wang An), a historically documented magistrate of Dongyang, as the most plausible referent. By integrating retrieved factual knowledge with contextual inference, the proposed framework enhances both the accuracy and consistency of annotations for ambiguous or indirect references.

Table 7. Examples of Inference for Chunk Referents.

No.	Classical Excerpt	PoemAnnotation Result
1	“怀薛尚书兼呈***东阳王使君***” *In Memory of Secretary Xue and Also Addressed to* ***Prefect Wang of Dongyang***	①“Refers to the Tang poet Li Jiayou, courtesy name Congyi, a native of Dongyang in Yue Prefecture. He served as governor of Hangzhou, Yuanzhou, and Taizhou. — ***Taiyan 2.0***” ②“‘Prefect Wang of Dongyang’ refers to a local official serving in Dongyang, who is a friend with whom the poet communicates or shares thoughts in the poem. — ***ChatGPT4o***” ③“‘Prefect Wang of Dongyang’: Refers to an official who once served as prefect of Dongyang, possibly referring to Wang An. —***Ours***”
2	“酬***张祜处士***见寄长句四韵” *In Reply to Recluse Zhang Hu’s Four-Rhyme Poem Sent to Me*	①“Zhang Hu, also known as Zhang Ji, courtesy name Wenchang, styled as 'recluse.' See Volume 6 of this book for biography. — ***Taiyan 2.0***” ②“‘Recluse Zhang Hu’ refers to the virtuous Zhang Hu, used here to address the person interacting with the poet, suggesting his talent and integrity made him worthy of respect. — ***ChatGPT4o***” ③“Refers to the Tang poet Zhang Hu, known for living in seclusion without holding official post, hence referred to as ‘recluse.’ Zhang Hu authored Two Palace Poems, one of which... According to《唐诗纪事》(*Records of Tang Poetry*), Zhang arrived in the capital and became affiliated with Yuan Zhen in the inner court... — ***Ours***”

For example 2, We observe that for the proper-name chunk “张祜处士” (“Zhang Hu the Recluse”), the Taiyan 2.0 model’s output depends critically on segmentation granularity: when provided only “张祜” as input, it correctly glosses “Zhang Hu, courtesy name Chengji, a poet esteemed by Linghu Chu,” but when given the full chunk “张祜处士,” it erroneously identifies “张祜” as Zhang Ji and fails to produce any accurate source citation. Similarly, ChatGPT-4o cannot generate precise biographical background or life details. In contrast, our model leverages a curated proper-name knowledge base to retrieve and integrate fine-grained information—such as Zhang Hu’s biographical history and exemplar works—thereby delivering more complete and accurate annotations.

5. Discussion

The automated annotation paradigm introduced in this paper overcomes the traditional reliance on manual chunk segmentation and the difficulty of lexical-sense disambiguation in prior approaches to classical-poetry annotation. Nevertheless, owing to the scarcity of high-quality, manually segmented corpora for classical-poetry chunking, there remains substantial room to improve segmentation performance. In future work, we will further enhance the retrieval efficiency and accuracy for out-of-vocabulary chunks by investigating more effective knowledge-base retrieval algorithms and semantic-matching techniques, thereby encompassing a broader spectrum of domain knowledge and advancing the digitalization and intelligent annotation of classical Chinese poetry. Moreover, the annotation framework proposed here establishes a foundation for downstream applications—including automated instructional support, difficulty estimation, and intelligent guided learning—in classical-literature pedagogy.

6. Conclusions

We propose a retrieval-augmented research paradigm for automatic annotation of classical Chinese poetry. By deeply integrating domain-specific dictionary knowledge, our method supplies precise, multi-layered glosses for a variety of semantic chunks. Compared to approaches that rely solely on general-purpose language models or purely generative techniques, this framework effectively mitigates semantic ambiguity and information loss when handling the diverse chunk types commonly found in poetry—such as allusions, imagery, and semantic shifts—thereby substantially improving annotation quality and controllability.

In implementation, for in-dictionary chunks (i.e., those with standardized entries in classical-poetry lexicons or knowledge bases), we first extract all candidate senses from the dictionary as initial glosses. These serve as reference anchors for the large language model's generation process, allowing the model to focus on contextual elaboration and stylistic refinement rather than constructing definitions from scratch. Comparison between the model's output and the initial dictionary glosses also enables rapid identification of discrepancies and targeted corrections, further enhancing annotation control.

For out-of-vocabulary chunks, we employ a two-stage retrieve-then-generate approach. First, we query external knowledge repositories or classical-poetry databases to retrieve semantically similar entries and relevant background information, thereby enriching the chunk's semantic context. Second, we supply these retrieved cues to the language model as auxiliary input, enabling it to produce accurate definitions for rare or domain-specific terms. This retrieval-augmented strategy not only increases the model's capacity to handle previously unseen chunks but also enhances the overall interpretability and robustness of the automatic annotation process.

Author Contributions: Conceptualization, W.Q. and J.L.; methodology, W.Q., D.W., and J.L.; software, J.L.; validation, T.W.; resources, X.X.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, W.Q., T.W., B.L. and M.F.; funding acquisition, W.Q. and D.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Major Project of the National Social Science Fund of China (No: 21&ZD288), and the Jiangsu Province Graduate Research and Practice Innovation Program project (No: KYCX22_1460).

Code and Data Availability Statement: Inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, J.; Wei, T.; Qu, W.; Li, B.; Feng, M.; Wang, D. Research on the Construction and Application of an Ancient Poetry Annotation Knowledge Base with Large Language Models. *Libr. Trib*. **2025**, 45, pp. 99–109.
2. Xu, Z.W.; Jain, S.; Kankanhalli, M. Hallucination is Inevitable: An Innate Limitation of Large Language Models. 2024. Available online: https://arxiv.org/abs/2401.11817.pdf (accessed on 3 November 2024).

3. Lewis, P.; Perez, E.; Piktus, A.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the 34th International Conference on Neural Information Processing Systems, BC, Canada, 06 December 2020.
4. Gushiwen. Available online: https://www.gushiwen.cn/ (accessed on 3 November 2024).
5. Gushici. Available online: https://shici.tqzw.net.cn/ (accessed on 3 November 2024).
6. Qu, T.; Zhu, J. *Li Bai Ji: Textual Collation and Annotation*; Shanghai Guji Publishing House: Shanghai, China, 1980.
7. Xiao, D.F. *Du Fu Quan Ji: Textual Collation and Annotation*; People's Literature Publishing House: Beijing, China, 2014.
8. Souyun. Available online: https://www.sou-yun.cn/ (accessed on 3 November 2024).
9. Shen, L.; Hu, R.; Wang, L. Construction and Application of Ancient Chinese Large Language Model. *Chin. J. Lang. Policy Plan*. **2024**, 5, pp. 22–33.
10. Hu, R.F.; Zhu, Y.C. Automatic Classification of Tang Poetry Themes. *Acta Sci. Nat. Univ. Pekinensis* **2015**, 51(2), pp. 262–268. https://doi.org/10.13209/j.0479-8023.2015.039.
11. Huang, Y.; Chen X.; Feng M.; et al. The Difficulty Classification of 'Three Hundred Tang Poems' Based On the Deep Processing Corpus. In Proceedings of the 22nd Chinese National Conference on Computational Linguistics. Harbin, China, 03-05 August 2023.
12. Liu, L.; He, B.; Sun, L. An Annotated Dataset for Ancient Chinese Poetry Readability. *J. Chin. Inf. Process*. **2020**, 34, pp. 9–18, 48.
13. Yao, R. An Automatic Analysis System for Poetry Based on the Ontology of Allusions. *Software Guide* **2011**, 10, pp. 80-82.
14. Tang, X.; Liang, S.; Zheng, J.; et al. Automatic Recognition of Allusions in Tang Poetry based on BERT. In 2019 Proceedings of International Conference on Asian Language Processing (IAlP). Shanghai, China, 15 – 17 November 2019.
15. Yi, X.; Sun, M.; Li, R.; et al. Chinese Poetry Generation with a Working Memory Model. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. Stockholm, Sweden. 13-19 July 2018.
16. Yi, X.; Li, R.; Yang, C., et al. MixPoet: Diverse Poetry Generation via Learning Controllable Mixed Latent Space. In Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, United States, 7-12 February 2020.
17. Bao, T.; Zhang, C. Extracting Chinese Information with ChatGPT: An Empirical Study by Three Typical Tasks. Data Anal. *Knowl. Discov.* **2023**, 7, pp. 1–11.
18. Yu, J. Chen, Feng, X. and Xia, Z. CHEAT: A Large-scale Dataset for Detecting CHatGPT-writtEn AbsTracts. *IEEE Transactions on Big Data* (Early Access) **2025**, pp. 1-9. https://doi.org/10.1109/TBDATA.2025.3536929.
19. Liu, Y.; Zhang, Z.; Zhang, W. ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models. *arXiv* 2023, arXiv: 2304.07666.
20. Bu, W.; Wang, H.; Li, X.; Zhou, S.; Deng, S. The Exploration of Ancient Poetry: A Decision-Level Fusion of Large Model Corrections for Allusion Citation Recognition Methods. *Sci. Inf. Res.* **2024**, pp. 37–52. https://doi.org/10.19809/j.cnki.kjqbyj.2024.04.004.
21. Cui, J.; Ning, M.; Li, Z.; et al. Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model. *arXiv* 2023, arXiv: 2306.16092.
22. LexiLaw. Available online: https://github.com/CSHaitao/LexiLaw (accessed on 3 November 2024).
23. Huang, Q.; Tao, M.; Zhang, C.; et al. Lawyer LLaMA Technical Report. *arXiv* 2024, arXiv: 2305.15062.
24. Xiong, H.; Wang, S.; Zhu, Y.; et al. DoctorGLM: Fine-tuning Your Chinese Doctor is Not a Herculean Task. *arXiv* 2023, arXiv: 2304.01097.
25. Wang, H.; Liu, C.; Xi, N.; et al. HuaTuo: Tuning LLaMA Model with Chinese Medical Knowledge. *arXiv* 2023, arXiv: 2304.06975.
26. XrayGLM. Available online: https://github.com/WangRongsheng/XrayGLM (accessed on 3 November 2024).
27. Liang, X.; Wang, H.; Zhao, Y.; et al. Controllable Text Generation for Large Language Models:A Survey. *arXiv* 2024 arXiv: 2408.12599.

28. Yue, S.; Chen, W.; Wang, Y.; et al. DISC-LawLLM: Fine-tuning Large Language Models for Intelligent Legal Services. *arXiv* 2024 arXiv: 2309.11325.
29. Setty, S.; Thakkar, H.; Lee, A.; et al. Improving Retrieval for RAG based Question Answering Models on Financial Documents. *arXiv* 2024 arXiv: 2404.07221.
30. Devlin, J.; Chang, M.; Lee, K.; et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, United States, 2-7 June 2019.
31. Ancient Chinese Corpus. Available online: https://catalog.ldc.upenn.edu/LDC2017T14 (accessed on 3 November 2024).
32. Xiao, S.; Liu, Z.; Zhang, P.; et al. C-Pack: Packed Resources For General Chinese Embeddings. *arXiv* 2023 arXiv: 2309.07597.
33. Zeng, A.; Xu, B.; Wang, B.; et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv* 2024 arXiv: 2406.12793.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.