QRGR: A Retrieval–Generation–Retrieval Framework for Quotation Recommendation with Large Language Models

Anonymous ACL submission

Abstract

Quotation recommendation aims to recommend an appropriate quotation for a given textual context, thereby enhancing rhetorical effect and semantic richness. However, Large Language Models (LLMs) still struggle with this task, often generating hallucinated or stylistically mismatched quotations. To address these challenges, we propose QRGR, a novel Retrieval–Generation–Retrieval framework that combines semantic retrieval with LLM-based generation. In the first retrieval stage, contextually relevant quotations are selected using a fine-tuned embedding model. These quotations serve as in-context demonstrations to guide the LLM in generating new candidate quotations. The generated candidates are then filtered and employed as semantic anchors in a second retrieval step, producing final recommendations that are both stylistically appropriate and source-grounded. To address the issue of the missing formal scene quotation dataset, we construct a new Chinese quotation dataset, featuring structured paragraphs and verified quotations from authoritative sources. Experiments on three datasets demonstrate that QRGR significantly outperforms strong retrieval-based and LLM-based baselines in both accuracy and contextual relevance, achieving new state-of-the-art performance. Our dataset and code are publicly available at https://anonymous.4open.science/r/QRGR-A-Retrieval-Generation-Retrieval-Framework-for-Quotation-Recommendation-DC35.

1 Introduction

Quotations, such as proverbs, aphorisms, and well-known sayings, enhance writing by conveying complex ideas concisely, strengthening arguments, and adding rhetorical appeal (Leith, 2011). Therefore, quotations are widely used in domains such as literary composition (Kronshage, 2013), social media posts (Kula and Grzelka, 2022), and official documents (Fetzer and Weizman, 2018).

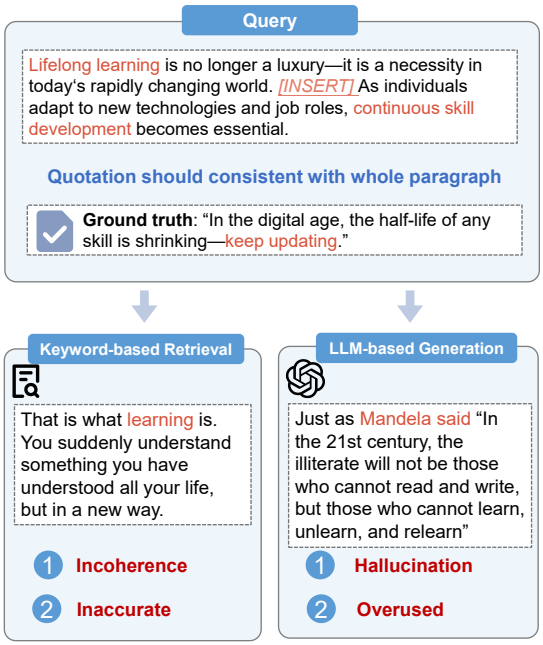


Figure 1: An example of the shortcomings of keyword-based retrieval and LLM-based generation methods. Keyword-based retrieval methods are only related to "learning", rather than "lifelong learning". The LLM-based method suffers from hallucination, generating quotations that are fabricated or from incorrect author, and often overused.

Despite their importance, selecting appropriate quotations remains a challenging task, as illustrated in Fig. 1. Quotations often differ from modern text in style, vocabulary, and structure, making them difficult to be recommended with keyword-based retrieval methods. Such methods typically return quotations that are topically related but semantically inaccurate, resulting in poor integration with the surrounding text. Large Language Models (LLMs), while capable of generating fluent and coherent content, frequently produce quotations that are not grounded in real sources (i.e., hallucination) (Ji et al., 2023). Besides, they tend to recycle

a limited set of formulaic expressions for diverse contexts. These issues are especially problematic in formal writing, where quotation accuracy, tone consistency, and contextual relevance are critical (Charteris-Black, 2011).

Recent methods for quotation recommendation have made notable progress by using pretrained language models (PLMs) to encode contextual semantics (MacLaughlin et al., 2021; Adak and Mukherjee, 2025). Additionally, sentence-level scoring functions, such as perplexity (PPL), have been applied to evaluate the fluency and contextual relevance of candidate quotations (Mao et al., 2023). LLM-based approaches (Xiao et al., 2024) further advance this line of work by designing reranking metrics and leveraging LLMs to assess how well candidate quotations fit within a given context. However, these methods typically treat generation and retrieval as two separate stages, failing to exploit the potential of retrieval results guiding generation, or vice versa.

In this paper, we propose a **R**etrieval-**G**eneration–**R**etrieval framework for **Q**uotation Recommendation (**QRGR**), integrating the strengths of both semantic retrieval and LLM generation. The framework begins by retrieving a small set of semantically relevant demonstrations using a fine-tuned embedding model. These demonstrations are used to guide an LLM to generate candidate quotations. Based on generated quotations, we perform a second retrieval step: the generated quotations are used to re-query the quotation knowledge base for the final selection.

While many quotation recommendation datasets (Qi et al., 2022; Wang et al., 2020) have been proposed, they typically limited in informal and loosely structured content from literary or daily conversational domains. Therefore, we introduce a new Chinese dataset that features structured paragraph-quote pairs, formal syntax, and domain-specific expressions such as official terms and proper nouns.

Our contributions can be summarized as follows:

(1) We propose QRGR, a retrieval–generation–retrieval framework for quotation recommendation based on the close collaboration between retrieval and generation.

(2) We construct and release a Chinese quotation dataset in the official document domain, consisting of 5,738 paragraphs and 53,638 unique quotations, featuring formal language and structured contextual alignment.

(3) We conduct extensive experiments on three datasets, demonstrating that our method consistently performs better across multiple evaluation metrics compared to competitive baseline methods.

2 Related Work

Quotation Recommendation. The task of quotation recommendation is originally introduced by Tan et al. (2015). They propose a learning ranking framework for this task, which integrates 16 hand-crafted features. They then introduced the LSTM (Graves, 2012) framework into the quotation recommendation task (Tan et al., 2016, 2018). Lee et al. (2016) propose a combination of multiple models, such as LSTM and CNN (Kim, 2014), to recommend quotations according to Twitter dialog threads. Wang et al. (2020) utilize a neural encoder-decoder framework to predict a quotation that continues the given conversation contexts. Wang et al. (2021) adopt a semantic matching approach which encodes the multi-turn dialog history with Transformer (Vaswani et al., 2017) and GRU (Cho et al., 2014). In the literary domain, RELic (Thai et al., 2022) introduced a dense retrieval model with a dual RoBERTa (Liu et al., 2019a) encoder framework, leveraging contrastive learning to optimize vector similarity between literary analysis contexts and correct citations. Subsequently, there are some methods that utilize PLMs to address quotation recommendation tasks. QuoteR (Qi et al., 2022) encodes position and semantic information based on BERT (Devlin et al., 2019). UniTrec (Mao et al., 2023) uses the Transformer encoder to model local- and global-attention, and introduces PPL to measure the relevance between users' and candidate text. QUILL (Xiao et al., 2024) was the first method to evaluate the ability of LLMs in the quotation generation task and designed a fine-grained quotation-specific metric to re-rank the retrieved quotations from the knowledge base.

Idiom Recommendation. A closely related task to quotation recommendation is idiom recommendation, which focuses on suggesting appropriate idiomatic expressions given a textual context. Existing idiom recommendation methods are essentially covered by the quotation recommendation methods. Liu et al. (2018) approach this task by leveraging word-level embeddings (Mikolov et al., 2013) to learn representations of both contexts and idioms, enabling the model to match semantically compatible pairs. Building on this, Liu et al. (2019b)

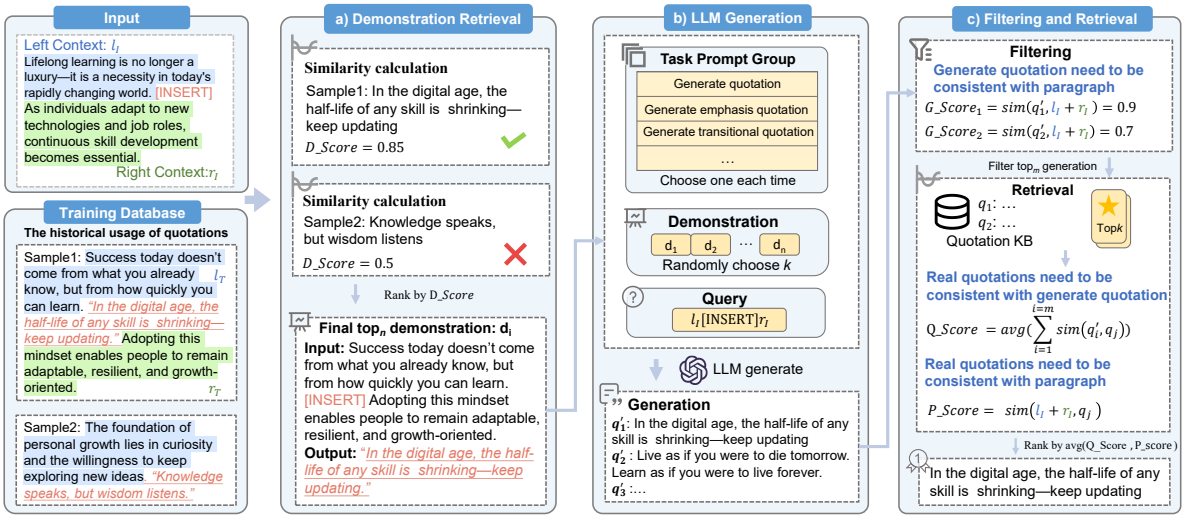


Figure 2: The overview of our QRGR framework. Given an input paragraph with a specified insertion point, we first retrieve quotations from the training database by computing similarity between the input paragraph and contexts paired with real quotations. Using these retrieved examples, we guide an LLM to generate candidate quotations for the given input paragraph. Finally, we filter the generated quotations and use the remaining ones to select an authentic quotation from the quotation knowledge base for the input paragraph.

reformulate idiom recommendation as a context-to-idiom machine translation problem, and implement an LSTM-based encoder-decoder framework to generate idioms conditioned on the surrounding context. Zhernokleev and Braslavski (2023) introduce an approach using fine-tuned Sentence-BERT (Reimers and Gurevych, 2019) models to recommend idioms by measuring semantic similarity between context and candidate expressions.

3 Retrieval-Generation-Retrieval Framework

This section presents our framework for quotation recommendation, as illustrated in Fig. 2. The framework comprises three main stages: (a) retrieving semantically related examples to construct demonstrations; (b) generating candidate quotations by inserting them at a special position within the input paragraph; and (c) obtaining the final recommendations through a retrieval process guided by the generated outputs.

We begin by formally defining the representation of inputs and training data. For an *Input*, we refer to the text before the insert position $[INSERT]$ as the left context l_I and the text after the $[INSERT]$ as the right context r_I. The left context and the right context together form an $Input = (l_I, r_I)$. For the instance t_i in the training dataset, the left context is defined as l_{t_i}, the right context is defined as r_{t_i}, and the quotation therein is defined as q_{t_i}. The collection of all these quotations, combined with additional quotations gathered from other sources, constitutes our quotation knowledge base, which serves as the candidate pool for the retrieval stage.

3.1 Contextual Retrieval

In our framework, the embedding model plays a vital role. The initial embedding model can distinguish between paragraphs and quotations, but it lacks the ability to align them semantically. Therefore, we fine-tune an embedding model to enable it to capture the correspondence between paragraphs and quotations.

We adopt BGE (Xiao et al., 2023) as the basic embedding model and further fine-tune it on the dataset of a specific domain. The training data for fine-tuning is structured in the form of triplets: $(Input, q^+, q^-)$, where q^+ denotes a positive quotation and q^- denotes a negative quotation. To enable BGE to not only complement the relationship between paragraphs and quotations but also distinguish differences among quotations, we use Jaccard Similarity to construct q^- by randomly sampling one of the top k quotations that is not used in the input paragraph from the quotation knowledge base.

Then, the same dataset used for fine-tuning BGE is also used to retrieve context-dependent demon-

strations for the LLM. The fine-tuned BGE is used to measure the similarities D_Score of the $Input = (l_I, r_I)$ and each $t_i = ((l_{t_i}, r_{t_i}), q_{t_i})$ to obtain context-dependent demonstrations for LLM generation, with the following equation:

$$\text{D_Score}(Input, t_i) = \frac{1}{2}[\cos(\mathbf{v}_{l_{t_i}}, \mathbf{v}_{l_I}) + \cos(\mathbf{v}_{r_{t_i}}, \mathbf{v}_{r_I})] \quad (1)$$

where $\mathbf{v}_{l_I}, \mathbf{v}_{r_I}$ denote the embeddings of the l_I and r_I, $\mathbf{v}_{l_{t_i}}, \mathbf{v}_{r_{t_i}}$ denote the embeddings of the l_{t_i} and r_{t_i}.

Based on these similarity scores, we retrieve the top n relevant paragraphs for the $Input$.

3.2 LLM-Based Generations

To address the gap between the input and authentic quotations, we employ an LLM to generate an imitative quotation, as shown in Fig. 2(b). Besides semantically coherently with the input, the quotation generated by LLMs are more stylistic and structural alignment with the target quotation.

To increase diversity and recall in the generated results, we design a prompt strategy group containing multiple rhetorical instructions. Each instruction corresponds to a specific function. For example, the model may be asked to insert a quotation that emphasizes a key point, introduces a shift in tone, or reinforces the preceding statement. This enables the model to explore different rhetorical roles for quotations within a paragraph, rather than producing uniformly styled outputs. The specific prompt strategies can be found in Appendix A.

To guide the generation process, we leverage the relevant paragraphs obtained in Section 3.1. Specifically, for each input, we randomly select k demonstrations from the top n retrieved quotations. Each demonstration consists of (l_{t_i}, r_{t_i}) with its real quotation q_{t_i}.

With the prompt constructed, we guide the LLM to generate N quotations for the $Input$, each generated quotation denoted as q'_i, and will be passed to the subsequent stage for final selection.

3.3 Filter and Retrieval

While the quotations generated by LLM are semantically relevant to the input, their quality varies greatly, due to the uncertainty of LLMs outputs. Therefore, the final stage of our framework consists of two components: (1) filtering the quotations generated by the LLM; and (2) retrieving the final quotation from the quotation knowledge base using both the generation and the input context as guidance.

3.3.1 Filter LLM-Generated Quotations

For each generated quotation q'_i, its generation score G_Score is calculated by the following equation:

$$\text{G_Score}(q'_i, Input) = \cos(\mathbf{v}_{q'_i}, \mathbf{v}_{Input}) \quad (2)$$

where $\mathbf{v}_{q'_i}$ and $\mathbf{v}_{Input}$ denote the embeddings of the generated quotation and the input paragraph.

We then retain the top m generated quotations with the highest G_Scores.

3.3.2 Final Retrieval from Quotation Database

To select the final quotation from the quotation knowledge base, we design a scoring mechanism that combines both generation-level alignment and context-level relevance. This strategy ensures that the selected quotation is not only stylistically aligned with LLM generation but also semantically appropriate for the input paragraph.

Generation-level score: For each authentic quotation q_j in the quotation knowledge base, its quotation score Q_Score is calculated with top m generated quotation by the following equation:

$$\text{Q_Score}(q_j) = \frac{1}{m}\sum_{i=1}^{m}\cos(\mathbf{v}_{q_j}, \mathbf{v}_{q'_i}) \quad (3)$$

Context-level score: For each authentic quotation q_j in the quotation knowledge base, its paragraph score P_Score is calculated with $Input$ by the following equation:

$$\text{P_Score}(q_j, Input) = \cos(\mathbf{v}_{q_j}, \mathbf{v}_{Input}) \quad (4)$$

Final score: The final retrieval score is a weighted combination of the two components:

$$\text{F_Score}(q_j) = \lambda \cdot \text{Q_Score}(q_j) + (1-\lambda) \cdot \text{P_Score}(q_j, Input) \quad (5)$$

where $\lambda \in [0, 1]$ is a hyperparameter that balances generation fluency and contextual relevance.

We rank all quotations in the quotation knowledge base based on *F_Score*, and select the top 1 quotation as the final recommendation.

Table 1: The statistics for the datasets. Left context and Right context columns denote whether the quotations in the dataset retain the preceding or following text when extracted.

Dataset	#Unique paragraphs	#Unique quotes	Left context	Right context	Avg #length for para	Avg #length for quotation
Reddit	44,539	1,111	Mutil-turn dialogue	None	1596.53	54.29
QuoteR	40,842	3,004	50 words	50 words	96.82	19.33
Official	5,738	53,638	whole paragraph	whole paragraph	206.08	32.95

4 Dataset Construction

To make up for the lack of datasets in formal and structured writing scenarios, we construct a new Chinese quotation dataset sourced from the official website [1], as shown in Tab 1. The Reddit dataset includes only the left context of quotations, while the QuoteR dataset provides both left and right contexts but extracts fixed-length snippets (50 words each) rather than complete paragraphs, thus failing to preserve the full discourse structure.

To ensure the quality and contextual relevance of the extracted quotations, we apply three key filtering strategies. (1) Named Entity Filtering: We retain only those paragraphs that mention at least one famous person, such as a political leaders, historical figures, scholars, or public intellectuals. (2) Length Constraint: We preserve quotations between 15 and 50 Chinese characters to filter out overly short fragments (e.g., idioms or interjections) as well as excessively long excerpts that may exceed sentence-level granularity. (3) LLM-Based Verification: Paragraphs meeting the above criteria are passed to an LLM to confirm the presence of a quotation. This final check ensures the accuracy and reliability of paragraph–quotation pairs (Han et al., 2023).

To better reflect practical application scenarios, where quotation recommendation involves selecting one item from a large number of candidates, we expanded our quotation knowledge base from the same official website. We collected additional quotation-only materials from designated speech archives on the website. After applying the length and LLM filtering strategy, these were used to construct a standalone quotation knowledge base for the retrieval tasks. Finally, we collect over 5,738 paragraph-quote pairs and 53,638 unique quotations.

[1] https://www.xuexi.cn/

5 Experiment

In this section, we conduct experiments to verify the effectiveness of our method.

5.1 Experiment Setup

Dataset To comprehensively evaluate our proposed framework across different contexts and styles, we conduct experiments on three quotation recommendation datasets.

- **Reddit Dataset**: Proposed by Wang et al. (2020), this dataset is constructed from Reddit and comprises multi-turn dialogs with corresponding idioms. The language used in this dataset is typically informal, conversational, and user-generated, making it suitable for assessing models in casual and dialogic settings.

- **QuoteR Dataset**: This benchmark dataset was introduced by Qi et al. (2022). The content is largely textual and sentence-level, and many entries originate from online sources, blending formal and semi-formal styles. This dataset is commonly used to evaluate quotation recommendation in general-purpose writing tasks.

- **Official Dataset**: Our official dataset includes official documents, speeches, and formal commentaries. This dataset reflects a highly formal and authoritative style, emphasizing rhetorical precision and contextual accuracy.

Each dataset is split into training, validation, and test sets with an 8:1:1 ratio.

Baselines We compare our framework against five representative baselines.

- Bi-GRU (Wang et al., 2021), a dual-encoder model that encodes queries and quotations into aligned semantic spaces using a transformation layer and an auxiliary mapping loss. This model was originally introduced with the Reddit dataset.

Table 2: Main results on all three datasets. All numbers are percentages (%). Best results are in bold, second best results are underlined.

Model	Reddit				QuoteR				Official			
	MRR	Hit@1	Hit@5	Hit@10	MRR	Hit@1	Hit@5	Hit@10	MRR	Hit@1	Hit@5	Hit@10
Bi-GRU	27.8	23.3	32.1	38.2	14.6	12.3	16.4	18.1	12.3	10.6	13.5	18.2
SememeBERT	39.7	31.4	48.4	55.7	54.1	47.9	60.1	65.0	27.2	18.1	36.8	46.1
UniTRec	35.1	30.2	41.5	48.5	41.2	38.1	52.9	61.9	20.1	12.5	25.7	41.1
QUILL	31.2	26.4	36.5	43.2	32.1	25.2	38.1	56.1	13.2	11.0	15.2	19.4
RA-MTR	33.1	27.5	37.9	44.6	49.8	45.7	59.2	66.1	29.5	20.6	39.4	47.3
QRGR (GLM4-9B)	39.1	29.8	49.7	54.6	29.5	22.4	36.9	43.1	27.5	16.4	38.6	50.1
QRGR (LLaMA3-8B)	42.8	**36.5**	51.2	57.6	47.9	39.2	58.2	63.8	24.2	11.3	38.4	52.2
QRGR (Qwen2.5-7B)	38.9	31.0	45.9	53.1	54.5	47.5	62.1	66.3	33.8	22.6	44.9	56.1
QRGR (Qwen2.5-14B)	39.8	31.5	46.7	52.6	54.1	46.9	62.8	67.3	39.5	27.9	52.3	62.5
QRGR (Qwen2.5-32B)	41.7	34.8	48.6	54.7	**59.7**	**53.5**	**66.4**	**69.9**	41.3	30.1	52.9	**66.1**
QRGR (Qwen3-8B)	40.3	33.0	47.2	54.3	52.8	45.9	60.2	64.2	37.3	26.9	48.3	59.3
QRGR (Qwen3-14B)	40.4	33.5	46.4	53.6	55.3	48.4	62.3	65.0	41.5	**31.9**	52.1	63.1
QRGR (Qwen3-32B)	**43.1**	35.6	**51.6**	**58.4**	56.7	50.3	63.8	67.0	**43.0**	31.6	**55.9**	65.7

- SememeBERT (Qi et al., 2022) combines a pretrained BERT (Devlin et al., 2019) encoder with sememe-level (Qin et al., 2020) lexical knowledge to enhance representation learning for short and abstract quotations. This method was introduced alongside the QuoteR dataset.

- UniTRec (Mao et al., 2023), a unified text-to-text Transformer framework with joint contrastive learning for text recommendation. It employs a local–global attention Transformer encoder to model both word-level and turn-level contexts in user history, leverages a Transformer decoder to estimate candidate text PPL, and unifies discriminative matching and PPL-based contrastive objectives to enhance recommendation performance.

- QUILL (Xiao et al., 2024), an LLM-based evaluation method that re-rank retrieved quotations using multiple PPL-based criteria, including quotation fluency, semantic coherence, and novelty.

- RA-MTR (Adak and Mukherjee, 2025), a retrieval-augmented multi-task reader framework designed for extracting contextually relevant inspirational quotations from long documents. It integrates a vector-store retriever with a Llama-3-based re-ranker, and a multi-task reader that performs quotable sequence tagging and context-aware span prediction.

Evaluation Metrics We use standard retrieval-based ranking metrics to evaluate performance, including mean reciprocal rank (MRR) and Hit@K at various cutoff levels (K=1, 5, 10). MRR captures how early the correct quotation appears in the ranked list, while Hit@K measures the proportion of cases where the ground-truth quotation is retrieved within the top K results.

5.2 Main Results

To evaluate the robustness of our framework, we adopt different LLMs as our backbone, such as LLaMA (AI@Meta, 2024), GLM (GLM et al., 2024) and Qwen (Team, 2024, 2025) series models of different ranges and sizes, as shown in Table 2. To better illustrate the influence of model parameters on experiments, we conduct comprehensive experiments on multiple variants of the Qwen series.

Performance of the QRGR Framework: Across all datasets and metrics, QRGR consistently outperforms previous PLM-based and LLM-based scoring methods. On the Official dataset, the best model Qwen3-32B, achieved MRR of 43.0 and 55.9 Hit@10, representing about 60% relative improvement over the previous state-of-the-art techniques.

Robustness Across Writing Styles: QRGR maintains a strong level of performance in informal conversations (Reddit), general writing (QuoteR), and official writing (Official). The performance of our framework on the Official dataset does not shown a significant decline, indicating that our QRGR framework is effective in handling formal and domain-specific writing.

Impact of Model Scale on Performance: Larger models offer richer semantic representations and stronger instruction-following capabilities. Interestingly, LLaMA3-8B achieves results on the Reddit dataset that are competitive with the much larger Qwen2.5-32B, suggesting that alignment with the dataset's language and domain can sometimes compensate for model size.

Table 3: Hyperparameter tuning results on Official. Best settings in bold.

#Gen	Top m	λ	MRR	Hit@1	Hit@10
1	1	0.5	39.1	28.9	62.6
5	1	0.5	40.1	30.1	62.6
10	**3**	**0.5**	**43.0**	**31.6**	**65.7**
15	3	0.5	42.2	30.5	64.1
10	5	0.5	41.3	29.8	62.0
10	7	0.5	40.2	28.5	61.4
10	3	0.1	31.1	17.6	58.7
10	3	0.3	35.8	22.1	63.0
10	3	0.7	41.9	31.2	64.2
10	3	0.9	39.7	28.3	63.3

5.3 Ablation Study

Hyperparameter Settings. We conduct controlled experiments to determine the optimal hyperparameter settings for the three-stage QRGR framework. The parameters include: (1)The number of generations per quotation in the LLM generation stage (2)The number m of retained quotations in the filtering stage (3)The weight coefficient λ used to combine paragraph–quote and generation–quote similarity scores in the final retrieval step.

Table 3 reports performance across different values for each hyperparameter on the Official dataset. We select the configuration that maximizes MRR and Hit@1. We found that generating too few quotations or retaining too few candidates reduces recall and weakens diversity. Conversely, excessive generation can lead to semantic redundancy and diminished overall precision due to the homogenization of candidate quotations. The coefficient λ balances the influence of paragraph–quote and generation–quote similarities. When λ places too much emphasis on paragraph–quote similarity, the final effect drops significantly because the recommended quotations only rely on the paragraphs themselves, similar to previous work using encoders.

Table 4: Component-wise ablation results on Official dataset.

Model Variant	**MRR**	**Hit@1**	**Hit@10**
Full QRGR	**41.3**	**30.1**	**66.1**
w/o First Retrieval	34.1	23.5	61.1
w/o First Retrieval & Gen	29.0	15.7	55.6
w/o Filtering	38.2	27.0	64.4
w/o BGE Fine-tuning	18.0	9.3	36.3

Component-wise Ablation. We conduct component-wise ablation study to evaluate the contribution of each module in our framework. As shown in Table 4, removing BGE fine-tuning leads to a notable drop in performance, indicating the importance of domain-adapted embedding models for capturing semantic similarity. Disabling either the first retrieval module or the second-generation stage also leads to a significant drop in performance, highlighting the complementary contribution of each stage to the final outcome. Removing the filtering stage slightly lowers overall performance, indicating its role in eliminating irrelevant candidates before retrieval. These results confirm that each component in our framework contributes to the overall effectiveness, and their combination is essential for achieving strong performance.

5.4 BGE embedding space

To better illustrate the relationship between paragraphs and quotations, as well as the underlying principle of our method, we analyzed the embedding distributions of paragraphs and quotations before and after fine-tuning, as shown in Fig. 4. We adopt Principal Component Analysis (PCA) for dimensionality reduction to visualize the embedding space.

Obviously, before fine-tuning, the vector space of quotations was aggregated, and paragraph embeddings are scattered and fragmented. The observed low similarity between quotations and their corresponding paragraphs suggests that the model lacks the ability to effectively align the two, although it can still distinguish among paragraphs and quotations individually.

After fine-tuning, paragraphs and quotations are distributed relatively evenly in the vector space, and the model has the ability to retrieve corresponding quotations based on paragraphs. Therefore, in the final retrieval, we take into account the paragraph scores instead of merely searching the quotation knowledge base based on the quotations generated by the LLM.

5.5 Case Study

To intuitively present the quotation recommendation process in our framework, we present illustrative case in Fig. 3.

The input paragraph centers on the theme of the "potential of every individual human" and requires a famous quotation by Martin Luther King. First, we directly searched in the quotation knowledge base using the paragraph's embedding. The retrieved sentences primarily pertained to "unity and cooperation" rather than the potential of each individual

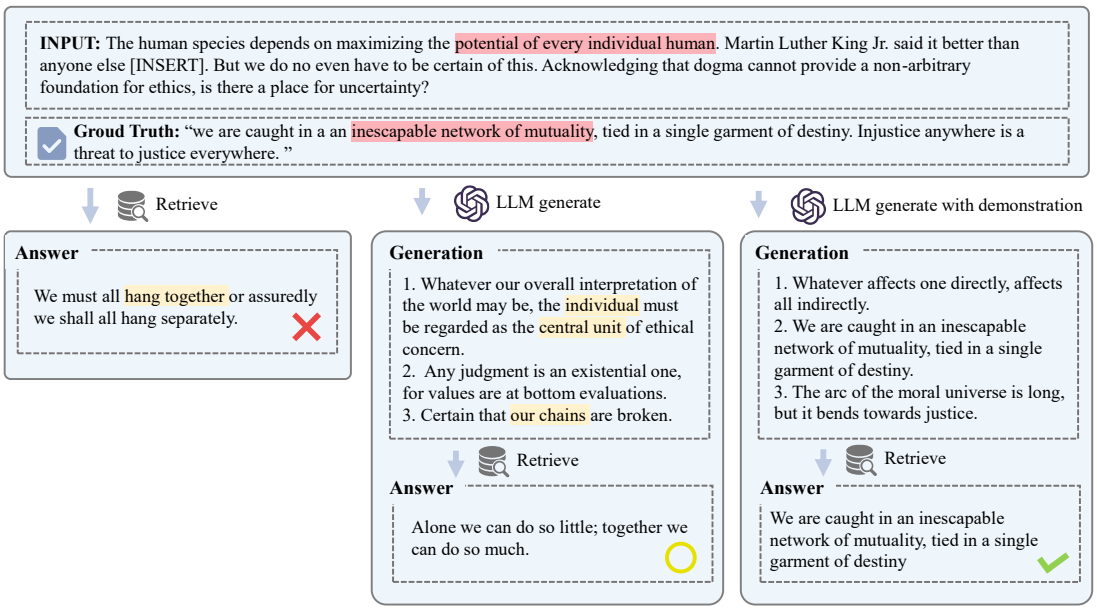
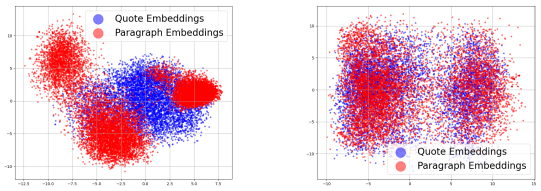


Figure 3: A case study on quotation recommendation. The answers retrieved using only paragraph semantic information do not match the ground-truth. The effect of retrieving the answers after generating them with an LLM has been improved to a certain extent. LLM with demonstrations have higher generation quality and match the ground-truth.



(a) Before fine-tuning (b) After fine-tuning

Figure 4: Visualization of embedding space before and after fine-tuning.

emphasized in the paragraph. This result highlights that relying solely on the semantic similarity of paragraphs, the obtained content usually do not align closely with ground-truth.

Then we asked the LLM to generate at the special position. The generated sentences related to themes such as "people-oriented" and "cooperation". Although these generated sentences provided helpful semantic cues to guide retrieval and partially matched the paragraph's intent, the LLM suffered from hallucination (Maynez et al., 2020) issue that cannot generate the corresponding author's famous sayings well. Consequently, the semantic alignment remained imperfect.

Finally, after supplying contextual demonstrations, the LLM better understood the paragraph's semantics and generated quotations accurately corresponding to the author's well-known sayings. Notably, the first two sentences were highly similar to the ground-truth. Leveraging these high-quality generated quotations, our framework successfully retrieved the appropriate quotes from the quotation knowledge base.

6 Conclusion

In this paper, we propose QRGR, a retrieval-generation-retrieval framework for quotation recommendation that tightly integrates semantic retrieval with LLM generation. By leveraging retrieval-guided demonstrations and iterative refinement, our method effectively mitigates hallucination and inconsistency issues commonly observed in LLM-based quotation recommendations. Experimental results on three datasets demonstrate that QRGR achieves state-of-the-art performance in both accuracy and contextual relevance. Our findings emphasize the value of combining contextual demonstrations with iterative retrieval to enhance quotation recommendation quality across both informal and formal writing domains.

Limitations

While the proposed QRGR framework advances the state of quotation recommendation, several limitations remain. Although the retrieval process in

QRGR helps improve the contextual consistency between quotations and paragraphs, the final recommendation quality still heavily depends on the coverage and diversity of the quotation knowledge base. In domains with sparse quotation resources, the model may still favor overused or generic quotations. Expanding and enriching quotation collections remains an important direction to improve recommendation diversity and depth. Meanwhile, our method still relies on domain-specific training data for fine-tuning the embedding model and constructing effective demonstrations. How to maintain high performance in scenarios with limited historical data or domain adaptation constraints remains a challenging open problem. Exploring zero-shot or few-shot learning strategies for quotation recommendation is a promising future avenue.

Another limitation concerns the evaluation design. The current dataset assigns only a single ground-truth quotation to each paragraph, which may not adequately reflect the inherent diversity of valid quotation choices. In practice, multiple quotations could be equally appropriate in terms of semantics and style; however, the existing evaluation protocol penalizes the model for recommending alternative yet contextually relevant quotations.

Ethics statement

The dataset used in this study was collected from Xuexi Qiangguo, a publicly accessible Chinese website. It consists solely of publicly available formal documents and contains no personally identifiable information. Our use of the dataset is strictly for academic research purposes and does not support, promote, or endorse any political or ideological views. This dataset provides a reliable and appropriate resource for studying formal Chinese texts.

References

Sayantan Adak and Animesh Mukherjee. 2025. Ra-mtr: A retrieval augmented multi-task reader based approach for inspirational quote extraction from long documents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5443–5462.

AI@Meta. 2024. Llama 3 model card.

Jonathan Charteris-Black. 2011. *Politicians and rhetoric: The persuasive power of metaphor*. Springer.

Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, volume 1 (long and short papers)*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, and 1 others. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Anita Fetzer and Elda Weizman. 2018. 'what i would say to john and everyone like john is...': The construction of ordinariness through quotations in mediated political discourse. *Discourse & Society*, 29(5):495–513.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and 37 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2023. An empirical study on information extraction using large language models. *arXiv preprint arXiv:2305.14450*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Eike Kronshage. 2013. The function of poetic epigraphs in george eliot's daniel deronda. *Connotations*, 23(2):230233.

Agnieszka M Kula and Monika Grzelka. 2022. Quotation in social media: How sharing other people's words could increase misinformation. *Studia Humanistyczne AGH*, 21(2):81–98.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 957–960.

Sam Leith. 2011. *You talkin'to me?: rhetoric from Aristotle to Obama*. Profile Books.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yuanchao Liu, Bingquan Liu, Lili Shan, and Xin Wang. 2018. Modelling context with neural networks for recommending idioms in essay writing. *Neurocomputing*, 275:2287–2293.

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019b. Neural-based chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5522–5526.

Ansel MacLaughlin, Tao Chen, Burcu Karagol Ayan, and Dan Roth. 2021. Context-based quotation recommendation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 397–408.

Zhiming Mao, Huimin Wang, Yiming Du, and Kam-Fai Wong. 2023. Unitrec: A unified text-to-text transformer and joint contrastive learning framework for text-based recommendation. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Fanchao Qi, Yanhui Yang, Jing Yi, Zhili Cheng, Zhiyuan Liu, and Maosong Sun. 2022. Quoter: A benchmark of quote recommendation for writing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 336–348.

Yujia Qin, Fanchao Qi, Sicong Ouyang, Zhiyuan Liu, Cheng Yang, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Improving sequence modeling ability of recurrent neural networks via sememes. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2364–2373.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jiwei Tan, Xiaojun Wan, Hui Liu, and Jianguo Xiao. 2018. Quoterec: Toward quote recommendation for writing. *ACM Transactions on Information Systems (TOIS)*, 36(3):1–36.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Learning to recommend quotes for writing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 65–74.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Katherine Thai, Yapei Chang, Kalpesh Krishna, and Mohit Iyyer. 2022. Relic: Retrieving evidence for literary claims. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7500–7518.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lingzhi Wang, Jing Li, Xingshan Zeng, Haisong Zhang, and Kam-Fai Wong. 2020. Continuity of topic, interaction, and query: Learning to quote in online conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6640–6650.

Lingzhi Wang, Xingshan Zeng, and Kam-Fai Wong. 2021. Quotation recommendation and interpretation based on transformation from queries to quotations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 754–758.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jin Xiao, Bowei Zhang, Qianyu He, Jiaqing Liang, Feng Wei, Jinglei Chen, Zujie Liang, Deqing Yang, and Yanghua Xiao. 2024. Quill: Quotation generation enhancement of large language models. *arXiv preprint arXiv:2411.03675*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Dmitrii Zhernokleev and Pavel Braslavski. 2023. Needle in a haystack: Finding suitable idioms based on text descriptions. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 185–196. Springer.

A Details of Prompt Strategy

Inspired by retrieval-augmented generation (RAG) (Lewis et al., 2020), our prompts incorporate retrieved context as input to guide the language model's response. We have disclosed our prompt group design, which can be seen in Tables 9–14.

B Prompt Strategy Comparison

We further evaluate the effect of different prompt strategies on the performance of the generation module. We compare our demonstration-based prompt (Few-shot) (Dong et al., 2022) with two alternatives: (1)Zero-shot prompt, which only provides task instructions without examples. (2)Chain-of-thought (CoT) prompt (Wei et al., 2022), which adds an intermediate reasoning step before generating the quotation.

Results in Table 5 show that few-shot demonstration prompt outperforms the other strategies. The zero-shot setting suffers from insufficient guidance. CoT prompts introduce intermediate reasoning, and the quality of the recommended quotations is similar to that without using CoT. To reduce time consumption, we did not use CoT.

Table 5: Prompt strategy comparison on Official.

Prompt Strategy	MRR	Hit@1	Hit@10
Zero-shot prompt	28.1	18.2	51.1
CoT prompt	32.2	21.4	54.8
Few-shot	**43.0**	**31.6**	**65.7**
CoT prompt & Few-shot	42.8	30.8	65.5

C Left and Right context of quotation

We performed ablation experiments on official dataset to evaluate the impact of contextual settings in our dataset design. Specifically, we compared performance under three conditions: (1) using only left context, (2) concatenating left and right contexts and treating them as a single left context, and (3) using our full dataset with explicit insertion positions. The results are shown in Table 6:

Table 6: Context setting on Official.

Context setting	MRR	Hit@1	Hit@10
Only left context	39.0	28.8	64.5
Left and right context connected as left context	40.9	29.7	64.5
Left and right context with insert position	**43.0**	**31.6**	**65.7**

D More Dataset Construction

We first collected review and news articles from the past 10–15 years as the initial dataset. To ensure data quality, we employed Qwen-7B to assist in the filtering process. Initially, a small subset of the crawled data was analyzed using Qwen-7B to identify the presence of celebrities within paragraphs, and the detected names were added to a curated celebrity list. After compiling approximately 100 celebrity names, we screened all collected paragraphs and retained those that (1) mentioned a celebrity, (2) contained quoted text enclosed in quotation marks, and (3) met the predefined length requirements. Finally, Qwen-7B was used to verify that each extracted quotation was indeed present within its corresponding paragraph.

E More Hyperparameter tuning

We provide more detailed results of hyperparameter tuning. Experimental findings show that variations in certain hyperparameters have only a marginal impact on model performance. Consequently, we streamlined parts of the experimental process and adopted the relatively optimal parameter configurations.

Table 7: Hyperparameter tuning results on Reddit. Best settings in bold.

#Gen	Top m	λ	MRR	Hit@1	Hit@10
5	3	0.5	40.1	32.1	55.8
10	3	0.5	42.6	34.2	57.1
15	3	0.5	42.4	34.0	57.5
10	5	0.5	42.8	34.7	58.0
10	3	0.5	43.0	34.9	**59.9**
10	**3**	**0.7**	**43.1**	**35.6**	58.4
10	3	0.9	42.5	34.5	57.5

Table 8: Hyperparameter tuning results on QuoteR. Best settings in bold.

#Gen	Top m	λ	MRR	Hit@1	Hit@10
5	3	0.5	54.9	49.3	65.3
10	3	0.5	55.3	49.7	66.1
10	5	0.5	55.1	49.8	65.9
10	3	0.3	53.3	47.5	64.2
10	**3**	**0.5**	**56.7**	50.3	**67.0**
10	3	0.7	56.1	**50.5**	66.3

For BGE fine-tuning, we adopted the default parameter settings. Specifically, we fine-tuned the bge-large-zh-v1.5 and bge-large-en-v1.5 models according to the respective languages using four NVIDIA 3090Ti GPUs. The learning rate was set to 1e-5, the warmup ratio to 0.1, and the number of training epochs to 2.

For the other baseline methods, experiments were also conducted on four 3090Ti GPUs, with all hyperparameters following the default configurations specified in the corresponding papers.

Prompt Example1
—SYSTEM PROMPT— You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please generate a contextually appropriate quotation at the location marked with [INSERT]. —OUTPUT REQUIREMENTS— 1.The quotation must be closely related to the surrounding context. 2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!! 3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples. If the input is highly similar to an example, you may reuse the corresponding output. —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —INPUT— {Input} —OUTPUT—

Table 9: Example of the common prompt.

Prompt Example 2
—SYSTEM PROMPT— You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please generate a contextually appropriate quotation at the location marked with [INSERT]. —OUTPUT REQUIREMENTS— 1.The quotation must be closely related to the surrounding context. 2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!! 3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples. If the input is highly similar to an example, you may reuse the corresponding output. 4.Please determine the relationship between the preceding and following text before and after [INSERT] and generate a quote that can express a transition. —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —INPUT— {Input} —OUTPUT—

Table 10: Example of the transition prompt.

Prompt Example 3
—SYSTEM PROMPT— You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please generate a contextually appropriate quotation at the location marked with [INSERT]. —OUTPUT REQUIREMENTS— 1.The quotation must be closely related to the surrounding context. 2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!! 3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples. If the input is highly similar to an example, you may reuse the corresponding output. 4.Please determine the relationship between the preceding and following text before and after [INSERT], select the more important part, and generate a sentence that emphasizes either the preceding or the following text. —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —EXAMPLE INPUT— {Left Text} [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —INPUT— {Input} —OUTPUT—

Table 11: Example of the emphasizes prompt.

Prompt Example 4
—SYSTEM PROMPT— You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please generate a contextually appropriate quotation at the location marked with [INSERT]. —OUTPUT REQUIREMENTS— 1.The quotation must be closely related to the surrounding context. 2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!! 3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples. If the input is highly similar to an example, you may reuse the corresponding output. 4.Please study the overall idea of the previous text and insert a quote that can summarize and generalize the idea in the [INSERT] section. —EXAMPLE INPUT— {Left Text} [INSERT] —EXAMPLE OUTPUT— {Real quotation} —EXAMPLE INPUT— {Left Text} [INSERT] —EXAMPLE OUTPUT— {Real quotation} —INPUT— {Input} —OUTPUT—

Table 12: Example of the prompt that only give left context.

Prompt Example 5
—SYSTEM PROMPT— You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please generate a contextually appropriate quotation at the location marked with [INSERT]. —OUTPUT REQUIREMENTS— 1.The quotation must be closely related to the surrounding context. 2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!! 3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples. If the input is highly similar to an example, you may reuse the corresponding output. 4.Please study the overall idea of the following text and insert a reference that can lead to the idea of the following text in the [INSERT] section. —EXAMPLE INPUT— [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —EXAMPLE INPUT— [INSERT] {Right text} —EXAMPLE OUTPUT— {Real quotation} —INPUT— {Input} —OUTPUT—

Table 13: Example of the prompt that only give right context.

Prompt Example 6

—SYSTEM PROMPT—
You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the text provided by the user, please recommend **topics** where sentences can be inserted at the position marked [INSERT].

—OUTPUT REQUIREMENTS—
1.The topics must be closely related to the surrounding context.
2.Output format: only the quotation in the form of: "Topic". Do not output anything beyond this required format!!
3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples.

—EXAMPLE INPUT—
{Left Text} [INSERT] {Right text}
—EXAMPLE OUTPUT—
{TopicA, TopicB, TopicC}

—EXAMPLE INPUT—
{Left Text} [INSERT] {Right text}
—EXAMPLE OUTPUT—
{TopicA, TopicB, TopicC}

—INPUT—
{Input}
—OUTPUT—

/**Each time, select one of the topics for following generation**/

—SYSTEM PROMPT—
You are a scholar well-versed in literary studies and familiar with famous quotes from both ancient and modern figures. Based on the provided text and topic, please generate a contextually appropriate quotation at the location marked with [INSERT].

—OUTPUT REQUIREMENTS—
1.The quotation must be closely related to the surrounding context and topic.
2.Output format: only the quotation in the form of: "quotation". Do not output anything beyond this required format!!
3.Before generating, carefully read and understand the surrounding text. Study the relationship between the input and output in the provided examples.

—EXAMPLE INPUT—
{Left Text} [INSERT] {Right text}
{Topic i}
—EXAMPLE OUTPUT—
{Real quotation}

—EXAMPLE INPUT—
{Left Text} [INSERT] {Right text}
{Topic j}
—EXAMPLE OUTPUT—
{Real quotation}

—INPUT—
{Input}
{Topic k}
—OUTPUT—

Table 14: Example of the CoT prompt.