HyReC: Exploring Hybrid-based Retriever for Chinese

Anonymous ACL submission

Abstract

Hybrid-based retrieval methods, which unify dense-vector and lexicon-based retrieval, have garnered considerable attention in the industry due to performance enhancement. However, despite their promising results, the application of these hybrid paradigms in Chinese retrieval contexts has remained largely underexplored. In this paper, we introduce HyReC, an innovative end-to-end optimization method tailored specifically for hybrid-based retrieval in Chinese. HyReC enhances performance by integrating the semantic union of terms into the representation model. Additionally, it features the Global-Local-Aware Encoder (GLAE) to promote consistent semantic sharing between lexicon-based and dense retrieval while minimizing the interference between them. To further refine alignment, we incorporate a Normalization Module (NM) that fosters mutual benefits between the retrieval approaches. Finally, we evaluate HyReC on the C-MTEB retrieval benchmark to demonstrate its effectiveness.

1 Introduction

Retrieval-augmented generation (RAG) enhances large language models by incorporating external knowledge to address hallucination issues, simultaneously catalyzing the rapidly evolving development of the retrieval community. How to effectively retrieve the most relevant information from the knowledge base is critically important for the final generation results. According to the encoding space, retrieval methods can be mainly categorized into three classifications: dense-vector(e.g., Condenser (Gao and Callan, 2021a), Bge embedding (Xiao et al., 2023), and Jina embedding (Sturua et al., 2024)), lexicon-based(e.g., DeepCT (Dai and Callan, 2019), SparTerm (Bai et al., 2020), and TILDE (Zhuang and Zuccon, 2021b)) and hybrid-based paradigms(e.g., COIL-full (Gao et al., 2021), Unifier (Shen et al., 2023) and Bge M3 (Chen et al., 2024)). Among them, the hybrid-based paradigm has garnered significant attention owing to its superior performance.

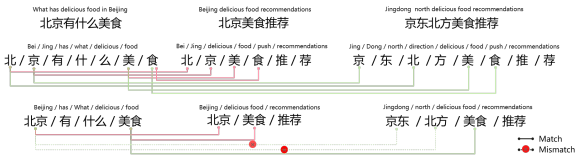

Figure 1: A example for lexicon-based retrieval. The three columns comprise the query, passage 1, and passage 2. The three rows illustrate the original text, the term-level matching results(terms derived from the tokenizer), and the word-level matching results(words generated by word segmentation), respectively.

The hybrid retrieval frameworks typically introduce a lexicon-based retrieval branch into the existing dense-vector model (Gao et al., 2021; Shen et al., 2023; Chen et al., 2024). The final matching score is computed as the sum of scores from both branches: the dense branch calculates similarity via the inner product of query and passage embeddings, while the lexicon-based branch is derived by multiplying the weights of the tokenizer-defined terms shared between the query and passage, followed by summing the resulting products. While this paradigm works adequately for English retrieval, it faces critical challenges in Chinese scenarios due to the absence of word boundaries (spaces). Specifically, lexicon-based matching relies on tokenizer-defined terms, which often fail to capture semantic nuances in Chinese. For instance, as illustrated in Fig. 1, term-level matching may incorrectly assign an identical score between Passage 1 and Passage 2, exposing semantic inconsistencies between term granularity and actual word meanings. This highlights the need for dedicated optimization of hybrid-based paradigms for Chinese.

It has been proven and widely accepted that in traditional lexicon-based retrieval, word-level matching properly can significantly improve performance. As illustrated in Fig. 1 (Row 3), such

improvements rely heavily on word segmentation modules to identify meaningful words. Widely adopted tools like Jieba[1] implement this through frequency-based heuristics, yet their lack of semantic awareness inevitably limits matching accuracy. To address this limitation, neural methods for word segmentation have emerged, utilizing bert-like models to capture semantic context (Tian et al., 2020; Huang et al., 2020; Maimaiti et al., 2021). Nevertheless, these approaches typically employ two separate models for words segmentation and lexicon-based retrieval, which lacks an end-to-end optimization solution and leaves room for performance enhancement.

In this paper, we present an innovative method called HyReC, which offers an end-to-end optimization solution for hybrid-based retrieval systems in Chinese scenarios. Specifically, HyReC integrates dense-vector retrieval, lexicon-based retrieval, and the semantic union of terms into a single model. The word segmentation is defined as the semantic union of terms to distinguish the difference between tokenizer-defined terms and model-defined semantic words. Within HyReC, the $[CLS]$ embedding is utilized for dense-vector retrieval, while embeddings from other tokens are employed for sparse retrieval and the semantic union of terms. During training, we have developed a labelling tool for training the semantic union, while the dense-vector and lexicon-based retrieval components are trained using a contrastive learning approach. Once trained, HyReC conducts large-scale retrieval either through its lexicon representation using an efficient inverted index or by leveraging dense vectors with parallelizable dot-product operations. In particular, each dimension of the lexicon representation corresponds to a term in the vocabulary, with its value reflecting the importance of that term within the passage. This vocabulary includes the result from the tokenizer's definition and the newly generated words generated by the semantic union of existing terms.

Moreover, we introduce an innovative module named the Global-Local-Aware Encoder (GLAE) to facilitate consistent semantic sharing, while simultaneously minimizing the interference between the two retrieval paradigms. Since the dense-vector paradigm is designed to learn sequence-level dense representations, the lexicon-based paradigm focuses on obtaining word-level lexicon representations (Shen et al., 2023). Additionally, we introduce a Normalization Module (NM) designed to align the two retrieval paradigms more reasonably, fostering mutual benefits. We normalize the matching scores of both paradigms to a 0-1 scale, rather than imposing rigid weights to enforce alignment (Chen et al., 2024).

Our main contributions of this paper are summarized as follows:

• We propose HyReC, a novel hybrid retrieval framework tailored for Chinese scenarios that unifies dense-vector retrieval, lexicon-based retrieval, and semantic union of terms within a single model.

• We additionally develop two key components: GLAE enables consistent semantic sharing while reducing paradigm interference, and NM achieves better alignment between two retrieval paradigms.

• Extensive experiments demonstrate that HyReC consistently outperformed baseline algorithms on the C-MTEB retrieval benchmark, validating its effectiveness.

2 Related Work

2.1 Dense-vector Retriever

To improve the retriever's performance, contemporary methods often focus on strategies such as selecting difficult negatives, leveraging pre-training and developing more elegant training recipes. For instance, ANCE (Xiong et al., 2020) introduced an innovative learning mechanism that globally selects difficult negatives from the entire corpus, utilizing an asynchronously updated approximate nearest neighbour (ANN) index. In contrast, ADORE (Zhan et al., 2021) employed dynamic sampling to adaptively adjust hard negative training samples during the model training process. Moreover, Condenser (Gao and Callan, 2021a) and coCondenser (Gao and Callan, 2021b) developed a pre-training strategy specifically designed for ad-hoc retrieval to enhance the performance of the model. Recently, Bge embedding (Xiao et al., 2023), and Jina embedding (Sturua et al., 2024) have introduced a three-stage training recipe and scaled training data to further enhance the retriever's effectiveness.

2.2 Lexicon-based Retriever

In recent years, researchers have been fervently working to enhance context representation in the lexicon-based retriever. DeepCT (Dai and Callan, 2019) translates contextual term representations

[1]https://github.com/fxsjy/jieba

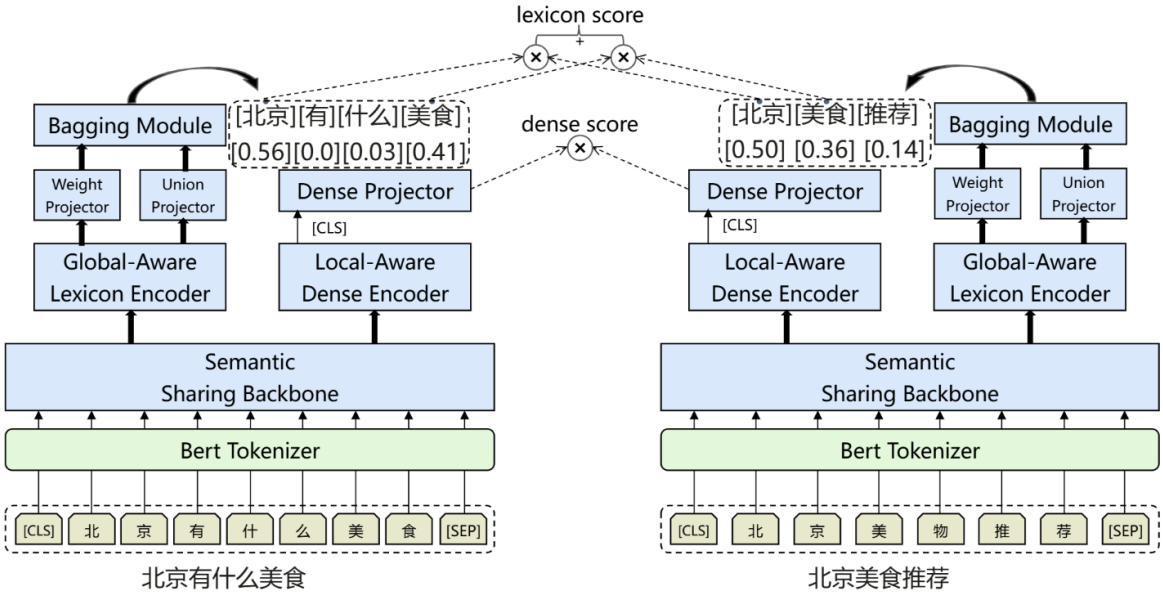

Figure 2: The architecture of our HyReC. HyReC first utilizes a semantic sharing backbone to extract low-level textual features for both paradigms. It then comprises two branches, each dedicated to learning global-aware and local-aware representations for the dense-vector retriever and the lexicon-based retriever, respectively. Additionally, a bagging module is employed to aggregate the weights and semantic union of terms, further enhancing the lexicon-based retriever's capabilities.

from BERT into term weights, deriving matching scores by multiplying the weights of terms shared between the query and passage and summing the resulting products. Similarly, SparTerm (Bai et al., 2020) introduced a contextual importance predictor that accurately assesses the significance of each term within the vocabulary. Building on contextual term representations, SPLADE (Formal et al., 2021) further introduced an innovative log-saturation effect that effectively regulates term dominance, promoting natural sparsity in the resulting representations. Additionally, TILDE (Zhuang and Zuccon, 2021b) proposed a more efficient framework for lexicon-based retrieval by incorporating a query likelihood component.

2.3 Hybrid-based Paradigms Retriever

The hybrid-based paradigm has garnered significant attention from the industry owing to its superior performance. COIL (Gao et al., 2021) used word-bag match and relied on $[CLS]$ vectors for computing relevance scores to assess hybrid-based retrievers. Bge M3 (Chen et al., 2024) expanded the ability of the hybrid-based retrieval by improving the training recipe and scaled training data. The authors in (Wang et al., 2021; Zhuang et al., 2024) explored normalization for combining dense and sparse retrievals. However, our method integrates normalization module during the training phase to jointly optimize the interaction between dense and sparse retrievals, rather than only applying it during inference. Unifier (Shen et al., 2023) integrates dense-vector and lexicon-based retrieval into a single model with dual representing capabilities. It also introduces a self-regularization method based on list-wise agreements from these dual views. However, to improve performance in lexicon-based retrieval, Unifier replaces the embedding of the $[CLS]$ token in the lexicon encoder with that from the local-aware dense encoder. This decision increases the inference time for lexicon-based retrieval and couples the lexicon-based retrieval with dense-vector retrieval, limiting flexibility in their applications. Additionally, the previously mentioned methods place limited emphasis on this scheme within the Chinese context and ignore the union between adjacent terms.

3 Methodology

HyReC seamlessly integrates dense-vector retrieval, lexicon-based retrieval, and the semantic union of terms into a single model. Additionally, it incorporates GLAE to ensure consistent semantic sharing while effectively minimizing interference between the two retrieval paradigms. Ultimately, it presents NM to align these two retrieval

approaches.

3.1 Network Architecture

As illustrated in Fig. 2, HyReC primarily consists of the semantic sharing backbone (detailed in Sec.3.1.1), the global-aware lexicon encoder (detailed in Sec.3.1.2), the local-aware dense encoder (detailed in Sec.3.1.3), three projectors (detailed in Sec.3.1.4) and the bagging module (detailed in Sec.3.1.4). The semantic sharing backbone, the global-aware lexicon encoder and the local-aware dense encoder are collectively referred to as GLAE.

3.1.1 Semantic Sharing Backbone

We begin by employing a semantic sharing backbone to extract low-level textual features for both retrieval paradigms, ensuring consistent semantic sharing. While the two paradigms concentrate on different levels of representation granularity(dense-vector retrieval focusing on sequence-level dense representation and lexicon-based retrieval emphasizing word-level contextualization embeddings), both paradigms delve into the semantic information of each term within the sentence. This shared exploration enables them to develop a cohesive understanding of semantic and syntactic knowledge directed toward the same retrieval targets. Like (Shen et al., 2023), we also leverage a multi-layer Transformer encoder to produce the semantic sharing backbone, i.e.,

$$S^{(x)} = TF\text{-}Enc([CLS]x[SEP]; \theta^{(ssb)}) \quad (1)$$

where $TF\text{-}Enc$ refers to a multi-layer Transformer encoder that utilizes parameters $\theta^{(ssb)}$. $[CLS]$ and $[SEP]$ are special tokens by following PLMs (Devlin et al., 2019). x represents either a query or a document.

3.1.2 Global-aware Lexicon Encoder

Building on the low-level textual features, we propose a representation module that generates a word-level lexicon representation. This module not only ensures consistent semantic sharing but also minimizes the interference from sequence-level dense representation associated with the dense retrieval paradigm. Unlike the approach taken in (Shen et al., 2023), we refrain from replacing the embedding of the $[CLS]$ token with that from the local-aware dense encoder for two key reasons: first, to reduce the inference time of the lexicon-based retrieval; and second, to decouple the lexicon-based retrieval and the dense-vector retrieval, allowing for more flexibility in their application. Given that the word-level lexicon representation captures global vocabulary space information, we designate this module as the global-aware lexicon encoder. To achieve this, we utilize an additional multi-layer Transformer encoder to process $S^{(x)}$. This can be expressed as

$$L^{(x)} = TF\text{-}Enc(S^{(x)}; \theta^{(gle)}) \quad (2)$$

where this module is parameterized by $\theta^{(gle)}$, which is distinct from $\theta^{(ssb)}$, the resulting $L^{(x)}$ denotes a word-level lexicon representation of the input text x, which is utilized for lexicon-based retrieval.

3.1.3 Local-aware Dense Encoder

Additionally, building on the low-level textual features, we present another representation module that generates a sequence-level dense representation. This module not only ensures consistent semantic sharing but also reduces interference from the word-level lexicon representation associated with the lexicon retrieval paradigm. Since the sequence-level dense representation does not incorporate global vocabulary space information, which captures local contextualization, we refer to this module as the local-aware dense encoder. To achieve this, we apply another multi-layer Transformer encoder to $S^{(x)}$, This can be written as

$$D^{(x)} = TF\text{-}Enc(S^{(x)}; \theta^{(lde)}) \quad (3)$$

where this module is parameterized $\theta^{(lde)}$, the resulting $D^{(x)}$ denotes a sequence-level dense representation of the input text x, which is employed for dense-vector retrieval. The dimension of $S^(x), L^(x)$ and $D^(x)$ is $[B, N, H]$, where B represents the batch size, N denotes the input sequence length, and H indicates the hidden size of the model.

3.1.4 Hybrid Retrieval

After obtaining the word-level lexicon representation $L^{(x)}$ and sequence-level dense representation $D^{(x)}$ of the input text x, we employ three projectors(i.e., weight projector, union projector and dense projector) to acquire the term weight, term union and dense vector, respectively.

To achieve term union, the union projector combines the word-level lexicon representation $L^{(x)}$ into four classification probabilities, indicating 'S' (single term word) 'B' (the beginning position of

the word), 'M' (the middle position of the word), and 'E' (the ending position of the word), respectively. This is expressed as:

$$U_{term_i} = softmax(w_u l_i + b_u) \quad (4)$$

where $term_i$ is the ith term or token in input x. w_u and b_u are linear weights and bias of the union projector module, respectively. l_i is ith token's word-level lexicon representations from $L^{(x)}$.

For term weight, we adopt a method inspired by the recent TILDEv2 (Zhuang and Zuccon, 2021a), which optimizes memory usage by storing only the scores of tokens that appear in current passages rather than the entire vocabulary. Differing from the original TILDEv2, our lexicon-based retrieval incorporates term union information to enhance performance. The weight projector integrates the word-level lexicon representation $L^{(x)}$ to produce a term importance score:

$$W_{term_i} = log(1 + ReLU(w_w l_i + b_w)) \quad (5)$$

where w_w and b_w are linear weights and bias of the weight projector module, respectively.

Lastly, the dense projector combines the representation of special token $[CLS]$ from the sequence-level dense representation $D^{(x)}$ to generate a sequence-level dense vector, which is utilized for dense-vector retrieval:

$$D_{vec} = (w_d d_{CLS} + b_d) \quad (6)$$

where w_d and b_d are linear weights and bias of the dense projector module, respectively. d_{CLS} is $[CLS]$ representations from $D^{(x)}$.

In the inference, we utilize the Bagging module to aggregate both the weights and the semantic unions of terms. To elaborate, we proceed as follows: first, based on U_{term_i}, we derive the result U_{word_j} for the jth semantic union. Notably, U_{word_j} may encompass multiple terms or tokens when U_{term_i} belongs to the set $\{B, M, E\}$; Second, we compute the weight W_{word_j} associated with U_{word_j}. as follows:

$$U_{word_j} = max(U_{term_i}); term_i \in word_j \quad (7)$$

$$W_{word_j} = max(W_{term_i}); term_i \in word_j \quad (8)$$

where $word_j$ represents a word that consists of more than one term or token.

The final matching score for hybrid retrieval is calculated as the sum of the matching scores from both the lexicon retrieval and the dense retrieval, i.e.,

$$S(q,p) = S^{lex}(q,p) + S^{den}(q,p) \quad (9)$$

where $S^{lex}(q,p)$ and $S^{den}(q,p)$ denote the matching score of the lexicon retrieval and the dense retrieval, respectively. The matching score of the lexicon-based method is calculated by weights multiplications of the common terms shared in the query and the passages.

$$S^{lex}(q,p) = \sum_{i,j} W^q_{\hat{term}_i} W^p_{\hat{term}_j} \quad (10)$$

where $W^q_{\hat{term}_i}$, $W^p_{\hat{term}_j}$ represents the weight of the ith term(or word) from the query and the passage, respectively. $\hat{term}_i$ is derived from both $term_i$ and $word_i$(similarly for $\hat{term}_j$.).

$$S^{den}(q,p) = D^q_{vec} \cdot D^p_{vec} \quad (11)$$

where D^q_{vec} and D^p_{vec} are sequence-level dense vectors of query and passage, respectively.

3.2 Loss

Given a query q and a set of n passages $D = \{p^+, \hat{p}_1, \hat{p}_2, ..., \hat{p}_{n-1}\}$. The lexicon retrieval and the dense retrieval task are acquired by the ranking objective with a contrastive loss. Thus, their training loss is

$$\mathcal{L}_* = -log \frac{e^{S^*(q,p^+)/\tau}}{e^{S^*(q,p^+))/\tau} + \sum_{\hat{P}} e^{S^*(q,\hat{p}_j)/\tau}} \quad (12)$$

where τ is the temperature parameter. $*$ is lex or den, which denotes the matching score of the lexicon retrieval and the dense retrieval, respectively. p and q^+ represent the paired texts, $\hat{p}_j \in \hat{P}$ denotes a hard negative.

The semantic union loss, represented as $\mathcal{L}_{union}$, is employed by the cross-entropy loss, as represented below.

$$\mathcal{L}_{union} = -\sum_i y_i log(U_{term_i}) \quad (13)$$

where y_i represents the labels of the term union (for more details, see Section 4.1.1).

The total loss for our HyReC is

$$\mathcal{L} = \mathcal{L}_{lex} + \mathcal{L}_{den} + \mathcal{L}_{union} \quad (14)$$

3.3 Normalization Module (NM)

In lexicon-based retrieval, matching scores are computed through weighted summation of identical terms, where the score range varies significantly depending on term weights. Similarly, dense retrieval scores obtained through vector dot products also exhibit unpredictable ranges. This discrepancy in score distributions makes direct combination problematic, necessitating normalization to align their scales. We first perform L2 normalization on the sequence-level dense representation $D^{(x)}$ before computing dot products. For the lexicon-based branch, we employ an attention mask to identify valid tokens, followed by L2 normalization of the term importance score vectors W_{term_i} for these selected tokens. The benefits of this module are as follows: First, it mitigates training instability caused by score disparities, which otherwise lead to inconsistent model preferences for dense or sparse retrieval across samples. Second, the normalization process effectively balances the contribution of each branch, allowing for more stable optimization. Finally, by aligning the score distributions, the model can learn more meaningful combination weights during training.

4 Experiments

4.1 Implementation Details

We follow the training recipe of BGE (Xiao et al., 2023) to ensure a fair comparison. See details of method and datasets in Appendix A.

4.1.1 Labelling Tool for Semantic Union

We combine the Jieba frequency word segmentation module with manually crafted regular expressions to generate labels for the semantic union. Since the Jieba framework relies primarily on frequency information and exhibits limited generalization, we enhance it by incorporating regular expressions to identify additional patterns, such as numeric expressions, quantity phrases, product models, and version identifiers. Labelling the semantic union involves the following steps: 1) Extracting the segmented words from the input and identifying the offsets of these segmented words within the original string; 2) Obtaining the tokenization results and their respective offsets; 3) Producing the labelling results by aligning the offsets between the segmented words and the tokenizer outputs. The offsets of the segmented words align a single token and the corresponding token is labelled as 'S' (indicating 0 class). For offsets that encompass multiple tokens, the tokens are labelled sequentially as 'B' (indicating 1 class), 'M' (indicating 2 class), and 'E' (indicating 3 class).

4.1.2 Evalution Metrics

In this paper, we explore the application of a hybrid-based paradigm within Chinese retrieval scenarios, focusing primarily on Chinese retrieval experiments. We adopt the C-MTEB (Xiao et al., 2023) retrieval benchmark as our standard, given its prominence in the field. Adhering to the official benchmark protocols, we evaluate our method using Pyserini (Lin et al., 2021) and utilize $nDCG@10$ as the primary evaluation metric.

4.1.3 Experimental Setups

For pre-training with a large volume of unsupervised data, we set the batch size to 512, with a maximum query and passage length of 512 tokens. The learning rate is configured at 1×10^{-4}, the warmup ratio is 0.1, and the weight decay is set to 0.01. This pre-training process is conducted across 16 V100 (32GB) GPUs.

In the two fine-tuning stages, we utilize a batch size of 64 for the small-scale model and 30 for the base-scale model. Additionally, we set the maximum lengths for queries and passages to 64 and 256 tokens, respectively. We perform 5 epochs with a learning rate of 5×10^{-5}, a temperature parameter of 0.05, and a weight decay of 0.01. The fine-tuning stage is executed on 8 3090 (24GB) GPUs. For the high-quality fine-tuning stage, we sample 3 negative instances for each query.

The small-scale model(with 38M parameters) is composed of a single-layer semantic sharing backbone, a single-layer global-aware lexicon encoder, and a single-layer local-aware dense encoder. In contrast, the base-scale model(with 153M parameters) incorporates a 5-layer semantic sharing backbone, a 7-layer global-aware lexicon encoder, and a 7-layer local-aware dense encoder. Notably, in small cases due to the constraints of our machine, we have opted not to scale our model to a large scale(like BAAI/bge-large-zh (Xiao et al., 2023)).

4.2 Main Evaluation

We conducted experiments using the C-MTEB retrieval benchmark (Xiao et al., 2023) to compare HyReC with the existing methods listed in Tab. 1. Our HyReC model exhibits remarkable advancements on the C-MTEB retrieval benchmark, It sig-

Model	T2	MM	Du	Covid	Cmed	Ecom	Med	Video	Avg
luotuo-bert-medium	58.67	55.31	59.36	55.48	18.04	40.48	29.8	38.04	44.4
text2vec-base-chinese	51.67	44.06	52.23	44.81	15.91	34.59	27.56	39.52	38.79
m3e-base	73.14	65.45	75.76	66.42	30.33	50.27	42.8	51.11	56.91
OpenAI	69.14	69.86	71.17	57.21	22.36	44.49	37.92	43.85	52.0
multilingual-e5-base	70.86	76.04	81.64	73.45	27.2	54.17	48.35	61.3	61.63
BAAI/bge-base-zh	83.35	**79.11**	86.02	72.07	**41.77**	63.53	56.64	**73.76**	69.53
BGE-m3-sparse	71.80	59.31	71.53	76.57	24.32	50.76	43.78	58.68	57.08
BGE-m3-dense	81.07	77.25	84.03	76.56	33.78	58.39	54.27	56.95	65.29
BGE-m3-hybrid	83.04	77.57	84.52	79.22	33.28	60.65	55.35	63.13	67.10
HyReC-base-sparse	73.00	69.43	74.58	74.41	30.76	59.64	48.29	67.07	62.15
HyReC-base-dense	82.93	77.40	**89.13**	76.06	34.42	61.31	57.30	72.16	68.84
HyReC-base-hybrid	**84.03**	77.81	87.31	**79.53**	38.47	**64.82**	**58.89**	73.46	**70.54**
multilingual-e5-small	71.39	73.17	81.35	72.82	24.38	53.56	44.84	58.09	59.95
BAAI/bge-small-zh	77.59	67.56	77.89	68.95	**35.18**	58.17	49.9	69.33	63.07
HyReC-small-sparse	73.20	68.23	74.67	75.93	28.67	59.30	46.64	68.48	61.89
HyReC-small-dense	76.90	70.30	82.76	72.43	33.81	55.04	51.02	66.05	63.54
HyReC-small-hybrid	**80.52**	**73.29**	**82.82**	**76.47**	34.93	**61.43**	**53.27**	**71.14**	**66.73**

Table 1: The experimental results on the C-MTEB retrieval benchmark are evaluated using $nDCG@10$. T2, MM, Du, Covid, Cmed, Ecom, Med and Video correspond to the T2Retrieval, MMarcoRetrieval, DuRetrieval, CovidRetrieval, CmedqaRetrieval, EcomRetrieval, MedicalRetrieval and VideoRetrieval development settings in C-MTEB retrieval benchmark.

Table 2: Ablation study on the effectiveness of each component on C-MTEB retrieval benchmark(SU means the semantic union of terms).

NM	GLAE	SU	Lexicon	Dense	Hybrid
	✓	✓	56.89	57.80	58.53
✓		✓	60.80	60.82	65.41
✓	✓		60.65	63.13	66.14
✓	✓	✓	**61.89**	**63.54**	**66.73**

nificantly outperforms Bge (Xiao et al., 2023) on $nDCG@10$ with a margin of +3.66%. A comparable improvement was observed when we scaled our model to a base size, resulting in an additional gain of +1.01% (notably, without the constraints of our machine, this improvement would be even more pronounced). Furthermore, the integration of lexicon-based and dense-vector retrieval leads to notable enhancements in retrieval performance, with improvements of +4.84% for lexicon-based retrieval and +3.19% for dense-vector retrieval. Ultimately, compared to the BGE-m3-hybrid approach which employs the same method (Chen et al., 2024), our hybrid-based retrieval method achieved a significant improvement in retrieval performance, reaching a 3.44% enhancement, demonstrating its outstanding effectiveness. Moreover, even without the inclusion of CovidRetrieval in the training data, our approach showcases its remarkable ability to generalize across diverse datasets.

4.3 Ablation Studies

4.3.1 Effectiveness of Each Component

The contributions of different components of HyReC are listed in Tab. 2. Utilizing the small-scale model of HyReC for this ablation study, we observed that the removal of the NM module leads to a significant drop in performance, highlighting the detrimental effects of an unpredictable score range on unstable training. The GLAE module was configured by adjusting the number of layers in the semantic sharing backbone, global-aware lexicon encoder, and local-aware dense encoder. The removal of the GLAE module entailed eliminating both the global-aware lexicon encoder and the local-aware dense encoder, while setting the number of layers in the semantic sharing backbone to two. The removal of the GLAE module leads to a decline in both lexicon-based (61.89% to 60.80%) and dense-vector (63.54% to 60.82%) retrieval performance, highlighting the interplay between the two retrieval paradigms. Furthermore, we conducted two additional experiments: 1) training exclusively on the lexicon-based retrieval task, which yielded an $nDCG@10$ score of 53.05%, and 2) training solely on the dense-vector retrieval task, resulting in an $nDCG@10$ score of

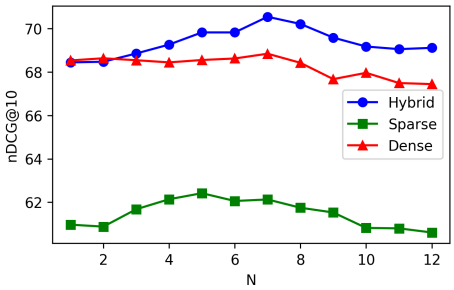

Figure 3: Ablation study on the performance of GLAE by the base-scale model, where the row axis N is the layer number of the global-aware lexicon encoder or the local-aware dense encoder and the layer number of the semantic sharing backbone is $12 - N$.

63.11%. Consequently, the semantic sharing backbone promotes consistent semantic sharing, as evidenced by substantial improvements—rising from 53.05% to 61.89% for lexicon-based retrieval and from 63.11% to 63.54% for dense-vector retrieval, with the most notable enhancement observed in lexicon-based retrieval. Additionally, introducing semantic union results in marked improvements: lexicon-based (rising from 60.65% to 61.89%), dense-vector (increasing from 63.13% to 63.54%), and hybrid-based retrieval (growing from 66.14% to 66.73%). This clearly illustrates that semantic union not only enhances lexicon-based retrieval but also positively affects dense-vector retrieval. The ablation studies presented in Tab. 2 verify the effectiveness of each module in our HyReC.

4.3.2 Parameters of GLAE

As depicted in Fig. 3, increasing the number of layers in the global-aware lexicon encoder or the local-aware dense encoder(while progressively reducing the layer number of the semantic sharing backbone) initially leads to an increase in all $nDCG@10$ scores. This initial enhancement can be attributed to a decrease in the interplay between the two retrieval paradigms. Beyond a certain point, the scores begin to decline, which is a consequence of reduced consistent semantic sharing.

4.4 Ablation of the Semantic Union of Terms

We conducted experiments to validate the proposed semantic union of terms. As illustrated in Tab. 3, the $nDCG@10$ performance of the semantic union surpasses the frequency-based method by 4.19%, 0.41% and 0.52% in lexicon-based, dense-vector and hybrid-based retrieval performance, respectively. These results clearly show that our proposed semantic union of terms outperforms the frequency-based method, i.e., Jieba. Additionally, we visualize the outputs of both the semantic union of terms and Jieba to assess the semantic information utilized in generating segmented words. As demonstrated in Tab. 4, the proposed semantic union of terms effectively comprehends the semantic nuances of the terms and achieves meaningful word segmentation by this semantic understanding.

Table 3: Ablation study on the performance of the semantic union of terms on C-MTEB retrieval benchmark.

Method	Lexicon	Dense	Hybrid
Jieba	57.70	63.13	66.21
HyReC	**61.89**	**63.54**	**66.73**

Table 4: The case study of the semantic union. Red words are incorrect segmentation.

Sentence	李一一一下子想不起她是谁 LiYiyi immediately couldn't remember who she was
Jieba	李/ 一一/ 一下子/ 想不起/ 她/ 是/ 谁 Li/ Yiyi/ immediately/ couldn't remember/ she/ was/ who
HyReC	李一一/ 一下子/ 想不起/ 她/ 是/ 谁 LiYiyi/ immediately/ couldn't remember/ she/ was/ who
Sentence	你告诉我光弱一端 You tell me weak light side
Jieba	你/ 告诉/ 我光弱/ 一端 You/ tell/ my weak light/ side
HyReC	你/ 告诉/ 我/ 光弱/ 一端 You/ tell/ me/ weak light/ side

5 Conclusion

In this paper, we introduced HyReC, an innovative end-to-end optimization method specifically designed for hybrid-based retrieval systems in the Chinese context. HyReC effectively integrates dense-vector retrieval, lexicon-based retrieval, and the semantic union of terms into a cohesive model to enhance overall performance. Additionally, our method incorporates two pivotal modules: (1) the Global-Local-Aware Encoder (GLAE), which facilitates consistent semantic sharing while minimizing interference between the retrieval paradigms, and (2) the Normalization Module (NM), which further fine-tunes the alignment between these retrieval paradigms. Our experimental results reveal that HyReC significantly outperforms the baseline, achieving remarkable improvements in $nDCG@10$ (+3.66% for the small-scale model and +1.01% for the base-scale model). The evaluations conducted on the C-MTEB retrieval benchmark conclusively demonstrate the effectiveness of our proposed approach.

6 Limitations

While our study presents a novel optimization module, semantic union of terms, tailored for enhancing retrieval tasks in Chinese retrieval scenarios, it is important to acknowledge two key limitations. First, the proposed module is specifically designed and optimized for Chinese language expressions, and its applicability to other languages remains unexplored. This limitation arises from the inherent linguistic characteristics embedded in the semantic union of terms, which are currently aligned with Chinese and may not directly generalize to multilingual contexts. Future work could investigate the adaptation of this module to other languages by incorporating cross-lingual semantic representations. Second, the experimental validation of our approach is confined to retrieval tasks on the C-MTEB benchmark, and its performance in other tasks, such as classification or clustering, has not been evaluated. This restriction stems from the fact that the semantic union of terms is inherently optimized for retrieval matching, and its effectiveness in broader applications remains an open question. Extending the evaluation to additional domains could provide a more comprehensive understanding of the module's versatility and potential impact. Additionally, due to the constraints of our GPU resources, we were unable to scale the model to a larger size(like BAAI/bge-large-zh (Xiao et al., 2023)) and the optimization of the base-scale model may not have been fully realized, limiting its potential performance.

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. mmarco: A multilingual version of the ms marco passage ranking dataset. *Preprint*, arXiv:2108.13897.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 6. long and short papers: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), 2-7 June 2019, Minneapolis, Minnesota, USA*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Michael Fuller, Marcin Kaszkiel, Sam Kimberley, Corinna Ng, Ross Wilkinson, Mingfang Wu, and Justin Zobel. 2008. 1 ad-hoc task 1.1 background.

Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Luyu Gao and Jamie Callan. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. *Preprint*, arXiv:1711.05073.

Kaiyu Huang, Degen Huang, Zhuang Liu, and Fengran Mo. 2020. A joint multiple criteria model in transfer learning for cross-domain Chinese word segmentation. pages 3873–3882, Online. Association for Computational Linguistics.

Jimmy J. Lin, Xueguang Ma, Sheng Chieh Lin, Jheng Hong Yang, Ronak Pradeep, Rodrigo Nogueira, and D. Cheriton. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-cpr: A multi domain chinese dataset for passage retrieval.

Mieradilijiang Maimaiti, Yang Liu, Yuanhang Zheng, Gang Chen, Kaiyu Huang, Ji Zhang, Huanbo Luan, and Maosong Sun. 2021. Segment, mask, and predict: Augmenting Chinese word segmentation with self-supervision. pages 2068–2077, Online and

Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Kai Zhang, and Daxin Jiang. 2023. Unifier: A unified retriever for large-scale retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4787–4799.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020. Improving Chinese word segmentation with wordhood memory networks. pages 8274–8285, Online. Association for Computational Linguistics.

Shuai Wang, Shengyao Zhuang, and G. Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. *Preprint*, arXiv:2205.12035.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwikj. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Sheng Zhang, Xin Zhang, Hui Wang, Lixiang Guo, and Shanshan Liu. 2018. Multi-scale attentive interaction networks for chinese medical question answer selection. *IEEE Access*, pages 1–1.

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2024. Promptreps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval.

Shengyao Zhuang and Guido Zuccon. 2021a. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513*.

Shengyao Zhuang and Guido Zuccon. 2021b. Tilde: Term independent likelihood model for passage reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1483–1492.

Model	T2	MM	Du	Covid	Cmed	Ecom	Med	Video	Avg
BAAI/bge-base-zh	82.38	**88.73**	87.27	86.78	**51.39**	**80.20**	65.90	86.00	78.58
BGE-m3-sparse	70.19	71.04	72.19	86.56	29.89	65.00	51.10	74.20	65.02
BGE-m3-dense	80.12	88.18	86.25	88.83	42.33	74.60	62.80	72.30	74.43
BGE-m3-hybrid	81.68	88.35	86.52	90.25	41.31	76.40	63.60	78.50	75.83
HyReC-base-sparse	72.28	80.62	76.06	87.20	37.10	74.40	55.70	82.90	70.78
HyReC-base-dense	81.70	87.93	**89.55**	88.41	42.59	76.30	66.30	85.40	77.27
HyReC-base-hybrid	**82.73**	87.83	88.28	**90.99**	46.83	79.40	**68.10**	**86.90**	**78.88**
BAAI/bge-small-zh	76.29	79.22	79.80	82.35	**44.08**	73.30	58.10	82.40	71.94
HyReC-small-sparse	72.45	79.77	75.74	88.09	34.91	72.80	54.60	83.80	70.27
HyReC-small-dense	75.94	81.45	83.69	84.56	42.60	70.60	60.40	81.00	72.53
HyReC-small-hybrid	**79.33**	**84.20**	**84.05**	**88.25**	42.43	**76.30**	**62.10**	**86.10**	**75.34**

Table 5: The experimental results on the C-MTEB retrieval benchmark are evaluated using $Recall$@10. T2, MM, Du, Covid, Cmed, Ecom, Med and Video correspond to the T2Retrieval, MMarcoRetrieval, DuRetrieval, CovidRetrieval, CmedqaRetrieval, EcomRetrieval, MedicalRetrieval and VideoRetrieval development settings in C-MTEB retrieval benchmark.

Model	T2	MM	Du	Covid	Cmed	Ecom	Med	Video	Avg
BAAI/bge-base-zh	92.13	74.52	90.95	70.77	**44.44**	59.32	53.62	67.85	69.20
HyReC-base-sparse	85.04	66.27	85.47	70.22	34.38	54.97	45.95	61.98	63.03
HyReC-base-dense	91.62	74.43	**94.77**	72.14	37.00	56.58	54.48	67.83	68.61
HyReC-base-hybrid	**92.47**	**74.92**	93.26	**75.89**	41.50	**60.24**	**56.05**	**69.08**	**70.43**
BAAI/bge-small-zh	88.47	64.27	86.59	64.66	38.04	53.39	47.32	65.15	63.49
HyReC-small-sparse	85.12	64.96	85.30	71.88	32.08	55.02	44.10	63.52	62.75
HyReC-small-dense	87.64	67.19	**90.70**	68.65	36.50	50.12	48.13	61.23	63.77
HyReC-small-hybrid	**90.29**	**70.19**	90.53	**72.68**	**38.26**	**56.78**	**50.53**	**66.30**	**66.94**

Table 6: The experimental results on the C-MTEB retrieval benchmark are evaluated using MRR@10.

A Training Recipe

Our training pipeline consists of both pre-training and fine-tuning phases. During fine-tuning, the global-aware lexicon encoder and local-aware dense encoder are initialized with the same pre-trained parameters. Specifically, we partition the pre-trained BERT model into two components: (1) the semantic-sharing backbone and (2) the encoder module. The latter is then used to initialize both the global and local encoders.

• **Pre-Training.** Utilizing the RetroMAE (Xiao et al., 2022) method, a variant of mask language modeling, we leverage the Wudao (Yuan et al., 2021) corpora to pre-train our model, which means we do not use any pre-trained language models.

• **Preliminary Fine-tuning.** At this stage, we gather text pairs from various open web sources, such as Zhihu and Baike. To enhance the quality of our dataset, we employ a third-party model, Text2Vec-Chinese[2], to filter out noisy data by applying a threshold of 0.43. Through this process, we successfully filter 160 million text pairs from the unlabeled corpora. Finally, the pre-trained model undergoes fine-tuning on this carefully curated corpus, which empowers it to effectively differentiate between the paired texts. Contrastive learning is employed to achieve local-aware dense representations and global-aware lexicon representations, while classification learning is utilized for semantic union. In-batch negative samples are adopted during training for contrastive learning.

• **High-quality Fine-tuning.** The model undergoes additional fine-tuning using a set of high-quality text pairs, which includes T^2-Ranking (Fuller et al., 2008), DURreader (He et al., 2018), mMARCO (Bonifacio et al., 2022), CMedQA-v2 (Zhang et al., 2018) and multi-cpr (Long et al., 2022). In total, there are 118,944,5 paired texts, most of which are curated through human annotation to ensure their high quality. During this stage, both contrastive learning and classification learning are employed to further refine

[2]https://huggingface.co/GanymedeNil

Model	T2	MM	Du	Covid	Cmed	Ecom	Med	Video	Avg
BAAI/bge-base-zh	76.02	73.93	76.98	70.80	**35.18**	59.32	53.52	67.85	64.20
HyReC-base-sparse	63.18	65.59	64.48	70.27	25.84	54.97	45.95	61.98	56.53
HyReC-base-dense	75.03	73.79	**82.58**	72.05	29.03	56.58	54.47	67.88	63.93
HyReC-base-hybrid	**76.32**	**74.39**	80.20	**75.82**	32.71	**60.24**	**55.99**	**69.08**	**65.60**
BAAI/bge-small-zh	68.59	63.60	68.39	64.64	29.24	53.39	47.27	65.17	57.54
HyReC-small-sparse	63.44	64.27	64.98	71.99	23.87	55.02	44.10	63.52	56.40
HyReC-small-dense	67.76	66.47	74.37	68.52	28.06	50.12	48.08	61.28	58.08
HyReC-small-hybrid	**72.01**	**69.58**	**74.44**	**72.65**	**29.49**	**56.78**	**50.48**	**66.30**	**61.47**

Table 7: The experimental results on the C-MTEB retrieval benchmark are evaluated using MAP@10.

the model. In contrastive learning, we not only utilize in-batch negative samples but also implement an ANN-style sampling strategy (Xiong et al., 2021) to generate hard negative samples. This stage features two key distinctions from the preliminary fine-tuning: firstly, it incorporates high-quality text pairs with human annotations for training; secondly, the negative sampling process is enhanced by the inclusion of hard negative samples.

B ANN-style Hard Negative Mining

Our ANN-style hard negative mining involves:

- building an index using the first-stage model.
- retrieving the top 100 passages per query.
- sampling hard negatives from non-positive passages ranked 20th–100th.

C Evaluation in Other Metrics

As illustrated in Tables 5-7, our retrieval method significantly enhances performance. This is evidenced by its impressive results across $Recall$@10, MRR@10, and MAP@10 metrics, highlighting its exceptional effectiveness(MRR@10 and MAP@10 metrics exclude the bge-m3 model due to the absence of pertinent evaluation codes on its official website.).