Search-R3: Unifying Reasoning and Embedding Generation in Large Language Models*

Yuntao Gui The Chinese University of Hong Kong Hong Kong SAR ytgui@cse.cuhk.edu.hk

Abstract

Despite their remarkable natural language understanding capabilities, Large Language Models (LLMs) have been underutilized for retrieval tasks. We present Search-R3, a novel framework that addresses this limitation by adapting LLMs to generate search embeddings as a direct output of their reasoning process. Our approach exploits LLMs' chain-of-thought capabilities, allowing them to produce more effective embeddings by reasoning step-by-step through complex semantic analyses. We implement this through three complementary mechanisms. (1) a supervised learning stage enables the model's ability to produce quality embeddings, (2) a reinforcement learning (RL) methodology that optimizes embedding generation alongside reasoning, and (3) a specialized RL environment that efficiently handles evolving embedding representations without requiring complete corpus re-encoding at each training iteration. Our extensive evaluations on diverse benchmarks demonstrate that Search-R3 significantly outperforms prior methods by unifying the reasoning and embedding generation processes. This integrated post-training approach represents a substantial advancement in handling complex knowledge-intensive tasks that require both sophisticated reasoning and effective information retrieval. Project page: https://github.com/ytgui/Search-R3

Keywords

Large Language Models, Reasoning Language Models, Sentence Embedding

1 Introduction

Large language models (LLMs) have transformed the landscape of natural language processing, demonstrating exceptional capabilities in text generation [6, 65, 74], problem-solving [70] and reasoning [12]. Among the key methodologies that enable modern LLMs to tackle intricate challenges is chain-of-thought (CoT) reasoning. This approach empowers models to decompose complex problems into manageable sequential steps, significantly enhancing their reasoning abilities [72]. CoT reasoning is typically activated by including explicit instructions such as "please think step-by-step" in prompts to the model. This simple directive transforms the model's behavior: rather than immediately generating a final answer, the model produces a detailed reasoning path that shows each intermediate step in its logical progression toward the solution. This transparent reasoning process not only improves performance on complex tasks but also enhances explainability, allowing users to understand the model's decision-making pathway.

James Cheng
The Chinese University of Hong Kong
Hong Kong SAR
jcheng@cse.cuhk.edu.hk

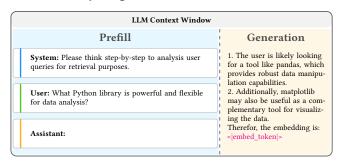


Figure 1: Illustration of Search-R3.

Despite these powerful reasoning capabilities, LLMs have been surprisingly underutilized in searching and embedding applications. Current approaches to search typically operate independently from LLMs and their reasoning processes, creating an artificial separation between how models comprehend content and how information is retrieved. This disconnection prevents the sophisticated reasoning capabilities of LLMs from enhancing search performance. In Retrieval-Augmented Generation (RAG) applications such as LlamaIndex [37], separate embedding models - typically BERT-based encoders like BGE [7, 13] - convert queries and documents into dense vectors for similarity retrieval, while the LLM only processes retrieved documents afterward in a disjointed pipeline. Recent advancement Search-R1 [26], which trains LLMs to generate better search queries during reasoning, still rely on external retrieval systems using either BM25-like text matching or embedding-based similarity that operate independently from the LLM's reasoning process, maintaining a fundamental disconnect between reasoning and retrieval. This separation between LLMs and embedding representation limits their ability to capture nuanced relationships between concepts, particularly in scenarios requiring intensive knowledge or multi-step reasoning.

We present Search-R3 (Reasoning-Reinforced Representation for Search), a novel framework that harnesses LLMs' reasoning capabilities to enhance embedding generation. Rather than treating embedding creation as an independent process, Search-R3 conceptualizes it as a direct outcome of analytical reasoning. Shown in Figure 1, our method leverages the standard LLM inference pattern of "prefill" and "generation" phases, where the prefill phase employs a carefully designed template containing system instructions for query analysis and the user's query itself. During the subsequent generation phase, Search-R3 produces two critical outputs sequentially: explicit analytical reasoning about the query's intent and identifies relevant concepts; and second, an embedding token <|embed_token|> that we've specifically trained Search-R3

^{*}This is a pre-publication draft. The copyright for this work belongs to the author(s). Please do not redistribute without permission.

to produce, which serves as a semantic representation encapsulating both the query and the analytical insights. Our comprehensive experiments across multiple benchmarks show that our approach delivers superior performance compared to existing methods.

The key innovations of our approach are:

- A novel embedding-through-reasoning architecture that enables LLMs to generate search embeddings as direct outputs of their analytical processes, fundamentally integrating semantic representation with explicit reasoning.
- A reinforcement learning framework that jointly optimizes reasoning processes and embedding outputs, creating a feedback loop where improved reasoning leads to more effective embeddings.
- A specialized RL environment that efficiently handles evolving embedding representations and makes the RL training computationally feasible for large-scale scenarios.

2 Background

In this part, we introduce the basics of information retrieval and LLM reasoning techniques to facilitate the subsequent discussions.

2.1 Revisiting Information Retrieval

Information retrieval has evolved from simple lexical matching to sophisticated semantic understanding. Classical approaches like TF-IDF [57, 61] and BM25 [55] operate on statistical word frequency patterns, calculating relevance scores based on term distribution properties. While computationally efficient and interpretable, these methods struggle with vocabulary mismatch problems and fail to capture semantic relationships between queries and documents when different terms are used to express similar concepts.

The limitations of lexical search have driven the advancement of dense retrieval systems, which encode text as continuous vectors in semantic space. Early Word2Vec [42] and GloVe [50] approaches capture semantic relationships between individual words, enabling tasks such as identifying similarity or resolving analogies, yet remain limited by their static, context-independent nature [5]. Transformer-based BERT models [13, 30, 38] later introduce contextual embeddings that capture meaning based on surrounding context, leading to more advanced sentence representation methods [15, 28, 54]. These approaches typically leverage contrastive learning paradigms that optimize similarity relationships, transfer learning mechanisms that adapt general language understanding to domain-specific tasks.

Despite these advances, existing embedding methods still struggle with complex semantic relationships that require deep conceptual understanding, lack the capacity for multi-step reasoning necessary for certain tasks, fail to construct explicit logical chains that connect ideas, and produce representations that remain largely uninterpretable to humans [43, 51, 56, 77].

2.2 Reasoning Mechanisms in LLMs

Large language models (LLMs) have advanced significantly in their reasoning abilities, transitioning from basic text completion to complex problem-solving through structured, step-by-step reasoning. This progress has been driven by the Chain-of-Thought (CoT)

methodology [72, 78]. By breaking down intricate problems into smaller and transparent steps, CoT enhances the accuracy and interpretability of LLM outputs.

Reinforcement learning (RL) has played a pivotal role in enabling the CoT capabilities of LLMs [12, 17, 65]. While supervised training can teach models to follow instructions, it often struggles to optimize CoT paths where multiple valid approaches exist with varying effectiveness [9, 53]. In Reinforcement Learning from Human Feedback (RLHF) [49], models are optimized using outcome quality on reasoning chains, improving logical coherence and reducing hallucinations in complex reasoning tasks. For mathematics, RL is used to optimize reasoning path of problem solving strategies [35, 60], demonstrating correctness improvements.

Most existing RL approaches employ Proximal Policy Optimization (PPO) [59] algorithm for training. PPO optimizes the LM generation policies by maximizing rewards toward higher-quality CoT reasoning paths. Group Relative Policy Optimization [60] (GRPO) has emerged as a particularly effective algorithm over PPO [12, 73], offering simplicites by computing reward advantage \hat{A} of PPO in a group-relative manner:

$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, r_2, ..., r_G)}{\text{std}(r_1, r_2, ..., r_G)}$$
(1)

where r_i represents the reward for the i-th response in a group of G responses sampled for the same question. This approach reduces computational requirements by eliminating the need for a separate value function model while improving performance on complex reasoning tasks.

2.3 The Disconnect Between Embedding and LLMs

A significant disconnect exists between embedding models and Large Language Models (LLMs), stemming from fundamentally different training objectives and architectural designs. Embedding models optimize for similarity metrics to create effective vector representations for retrieval tasks, while LLMs are trained through next-token prediction to generate coherent and contextually appropriate text sequences.

This divergence in training objectives has led to distinct architectural approaches. Embedding models have predominantly followed the BERT-based encoder-only architecture [7, 34, 54, 69], which processes entire input sequences simultaneously to produce fixed-length vector representations. In contrast, LLMs typically employ decoder-only architecture [6] that process text autoregressively, building representations that evolve with each generated token.

Recent efforts have begun to bridge this gap by fine-tuning LLMs for embedding tasks [62, 77]. These approaches either adapt LLMs specifically for embedding generation (often sacrificing their instruction-following capabilities in the process) or simply extract embeddings from model outputs without utilizing the sophisticated reasoning capabilities of LLMs. In both cases, the embedding generation remains disconnected from the generative processes, e.g., reasoning, that give LLMs their power, treating embedding as a separate function rather than an integrated aspect of language understanding.

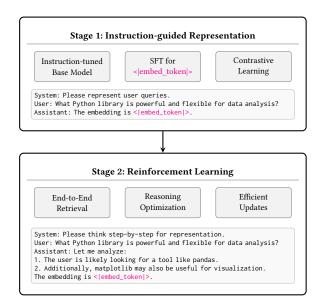


Figure 2: Training pipeline of Search-R3.

3 Overview

Our framework transforms an instruction-tuned base model (e.g., Qwen) into powerful embedding generators through a systematic two-stage training pipeline. The first stage integrates supervised fine-tuning (SFT) with contrastive learning (Section §4.1), teaching the model to recognize and respond to our specialized <|embed_token|> token while developing embedding generation capabilities within conversational contexts. This instruction-based representation learning utilizes the model's existing instruction-following mechanisms to produce embeddings in response to user queries, functioning similarly to its standard conversational generation processes.

The second stage employs reinforcement learning (RL) to optimize embedding quality in an end-to-end retrieval environment (Section §4.2). This phase enhances the model's ability to generate more effective embeddings by optimizing the intermediate reasoning process: the RL environment encourages the model to produce useful step-by-step semantic analyses before generating the final embedding. Additionally, we introduce an efficient RL environment design (Section §5) that manages evolving embedding representations without re-encoding of the entire corpus at each iteration, substantially reducing computational demands while preserving training effectiveness.

4 Methodology

This section details the algorithms and implementation that enable embeddings to emerge directly from analytical reasoning processes.

4.1 Instruction-guided Representation

The first stage addresses a fundamental challenge in leveraging

language models for embedding generation: LLMs are inherently optimized for next-token prediction within sequences rather than producing fixed-dimensional semantic vectors that capture meaning in a compressed form. We bridge this gap by introducing a special embedding token <|embed_token|> into the model's response, allowing us to extract embeddings directly from the model's hidden states. Notably, our approach maintains the exact architecture of the base model without introducing any additional components such as projection layers or dedicated embedding heads. This preserves the model's original parameter structure while enabling an entirely new capability, so that our method is orthogonal to all existing LLM inferencing tools, frameworks, and optimization techniques.

Building upon this architectural design, our approach reuses the conversation-based interface of LLMs through a three-part prompt structure:

System: Please representation user queries.

Assistant: The embedding is < embed_token >.

This conversational format creates a natural context for embedding representation that aligns with the instruction-following capabilities the base model already possess. When the model processes this structured input, we extract the hidden activation state from the final Transformer layer at the position of <|embed_token|>. This yields a fixed-dimensional vector $h \in \mathbb{R}^d$ that serves as our semantic embedding for the input text, where dimension d corresponds to the model's native hidden state size.

To optimize the embedding generation, we employ a composite loss function:

$$L = L_{SFT} + L_{KL} + L_{InfoNCE} + L_{TripletMargin}$$
 (3)

The $L_{\rm SFT}$ component implements the standard cross-entropy loss for language modeling [6, 49, 66], ensuring the model consistently produces the expected response format with the embedding token at the appropriate position. The $L_{\rm KL}$ component applies Kullback-Leibler divergence [18, 21, 33] to minimize distribution shifts between the fine-tuned model and the base model, preserving the model's general language understanding capabilities.

The InfoNCE contrastive loss [48] $L_{InfoNCE}$ is formulated as:

$$L_{\text{InfoNCE}} = -\log \frac{\exp(\cos(h_q, h_d^+)/\tau)}{\sum_{i=1}^K \exp(\cos(h_q, h_d^i)/\tau)}$$
(4)

where h_q represents the query embedding, h_d^+ represents the positive document embedding, $\sum_{i=1}^K$ is the cumulation of all K document embeddings, and τ is a temperature parameter. This loss structures the embedding space to cluster semantically similar items together while distancing dissimilar ones, teaching the model to encode semantic relationships in its vector space representations. Following existing successful training approaches in contrastive learning [15, 52], we set $\tau=0.05$, which provides well-separated clusters in the embedding space.

The triplet margin loss [4, 54] $L_{\text{TripletMargin}}$ is defined as:

$$L_{\text{TripletMargin}} = \max(0, \cos(h_q, h_p) - \cos(h_q, h_n) + \theta)$$
 (5)

where the query h_q serves as the anchor, θ is the margin parameter, h_p and h_n are positive and negative embeddings, respectively. This loss further refines the embedding space by enforcing explicit

Your task is to enrich user input for more effective embedding representation by adding semantic depth. For each user input, please think step-by-step briefly to:

1. Identifying core concepts and their relationships.

2. Including key terminology with essential definitions.

3. Adding contextually relevant synonyms and related terms.

4. Connecting to related topics and common applications.

After the analytical content, you MUST end every response with sembed token semantic contents.

Figure 3: System prompt in Stage 2.

distance constraints, ensuring positive documents remain closer to the query than negative ones by at least the margin θ , we set the margin parameter $\theta=0.15$ by pactice.

Through this comprehensive training approach, we effectively transform the LLM's next-token prediction capability into a mechanism for generating high-quality semantic vectors. The model learns to analyze input text before producing an embedding that encapsulates its semantic essence. This first stage establishes the foundation for the subsequent reinforcement learning phase: without this initial training, the model would lack the ability to reliably generate the embedding token required for embedding extraction and reward calculation in Stage 2.

4.2 Reinforcement Learning

While Stage 1 establishes the foundation for embedding generation, the resulting embeddings are primarily optimized for adhering to the supervised training dataset rather than for end-to-end retrieval performance. To address this limitation, we implement a reinforcement learning framework that directly optimizes both the reasoning process and the embedding quality.

Stage 2 employs the similar structured conversation format as Stage 1 with a system prompt designed to elicit richer semantic analysis, shown in Figure 3. We maintain the structural requirement from Stage 1: responses must end with the embedding token. We then optimize the quality of the reasoning path that produces the embedding.

We employ GRPO with a carefully designed reward function that enforces both structural compliance and retrieval quality:

$$R(q,r) = \begin{cases} -1.0, & \text{if no } <|\text{embed_token}| > \text{in } r \\ \text{DCG}_{\text{scaled}}(E(q,r),C), & \text{otherwise} \end{cases}$$

Here, q represents the input query, r denotes the model's generated response, E(q,r) extracts the embedding vector from the position of <|embed_token|>, and C is the retrieval corpus. The reward function strongly penalizes responses that fail to include the embedding token with a fixed negative reward of -1.0, ensuring the model learns to consistently produce embeddings. For responses containing the embedding token, we compute a scaled Discounted Cumulative Gain (DCG) that evaluates retrieval quality:

$$DCG_{scaled} = S \cdot \sum_{k=1}^{K} \frac{(P_k - 0.5 \cdot N_k)}{1 + \log(k)}$$
 (7)

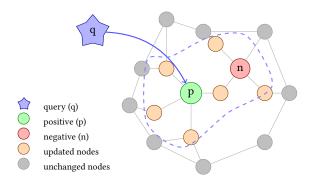


Figure 4: Illustration of selective graph refresh mechanism.

In this equation, k indexes the rank position from 1 to K (typically K=100), S_k represents the cosine similarity between the query embedding and document at rank k (ranging from -1 to +1), P_k equals 1 for positive matches and 0 otherwise, while N_k equals 1 for negative matches and 0 otherwise. The denominator $1 + \log(k)$ serves as a rank-based discount factor that prioritizes higher ranks.

The DCG_{scaled} function design provides several complementary signals. The DCG component encourages effective discrimination, rewarding the retrieval of positive content at the top ranks while penalizing negative retrievals at a 2:1 ratio. The cosine similarity term *S* introduces fine-grained scoring even when rank positions become stable, as models mature and consistently achieve top-1 rank retrievals, this similarity measure prevents reward saturation by rewarding closer embeddings.

During training, we generate multiple reasoning paths per query using higher sampling temperature ($\tau=1.2$). For each query q, we sample G=16 different responses, creating a diverse group of candidate reasoning paths and embeddings. We then compute advantages using the GRPO formulation across the group, enabling the model to learn which reasoning strategies produce more effective embeddings. Additionally, we incorporate a curriculum learning approach that gradually increases the difficulty of retrieval tasks. We begin with a small corpus of 65536 documents, progressively scaling up to 1 million document. This approach allows the model to first master retrieval in a less challenging environment before tackling increasingly complex retrieval scenarios with more potential distractors.

5 Scalable Reinforcement Learning Environment

Optimizing embeddings through reinforcement learning at production scale presents significant computational challenges, particularly when reward signals require evaluating retrieval performance, and the embedding space continually evolves during training. We address these challenges through a novel environment design that efficiently handles these evolving representations without requiring complete corpus re-encoding at each training iteration, otherwise, this would pose a prohibitive computational bottleneck.

Dataset structure. The RL environment is built upon datasets

Algorithm 1 Localized Graph Refresh

```
1: procedure RefreshGraph(queries Q, positives P, negatives
     N, embedding model \phi_t, graph G_{t-1})
           // Step 1: Find k-nearest neighbors
 2:
           \mathcal{N}_P \leftarrow \text{kNN}(P, G_{t-1}, k)
 3:
           \mathcal{N}_N \leftarrow \text{kNN}(N, G_{t-1}, k)
 4:
           // Step 2: Expand to 2-hop neighborhoods
 5:
           \mathcal{N}_{p}^{2hop} \leftarrow \text{Expand}(\mathcal{N}_{P}, G_{t-1})
 6:
          \mathcal{N}_{N}^{2hop} \leftarrow \operatorname{Expand}(\mathcal{N}_{N}, G_{t-1})
 7:
           \mathcal{N}_{combined} \leftarrow \mathcal{N}_{p}^{2hop} \cup \mathcal{N}_{N}^{2hop}
 8:
           // Step 3: Batch update of selected neighborhoods
 9:
           \mathcal{D} \leftarrow \text{GetDocuments}(\mathcal{N}_{combined})
10:
           \mathcal{H}_{new} \leftarrow \text{Batched embedding } \phi_t(\mathcal{D})
11:
12:
           // Step 4: Update graph structure with local join operation
13:
           G_t \leftarrow \text{LocalJoinUpdate}(G_{t-1}, \mathcal{N}_{combined}, \mathcal{H}_{new})
           return G_t
14:
15: end procedure
```

organized as query-positive-negative triplets:

$$T = (q_i, p_i, n_i) \mid q_i \in Q, p_i \in D, n_i \subset D$$
 (8)

Each triplet consists of a query q_i from the query set Q, a positive example p_i from the document corpus D that is semantically relevant to the query, and a hard negative example n_i from the same corpus D that contains subtle but significant factual differences to the query. This triplet structure provides the basis to perform end-to-end embedding evaluation: we present a query to the model to generate a query embedding, perform embedding search to obtain top-k results, then check if the positive and negative documents p_i and n_i appears within these k items. The position of p_i and n_i in the results directly determines the quality of the embedding and the corresponding reward signal, see Section §4.2.

Asymmetric generation. We apply an asymmetric strategy to generate the query and document embeddings. For documents D, we use a fixed conversation template that pair the corpus with an embedding token, performing a single-pass forward computation to generate the embedding. This allows the document corpus to be pre-encoded to embeddings (which we selectively update during training, as detailed in later), where performing autoregressive generation for the entire corpus would be computationally prohibitive. For queries Q, the conversation template is open-ended, enabling the model's step-by-step reasoning through autoregressive generation before finally producing the query embedding. The query-side generative process enables deeper semantic analysis to capture nuanced user intent.

Evolving search graph. The core of our rewarding infrastructure is a graph-based embedding search system using Hierarchical Navigable Small World (HNSW) index:

$$G_t = (V_t, E_t, \omega_t, \phi_t) \tag{9}$$

At training step t, our time-varying graph G_t consists of:

- $V_t = v_i^t \mid d_i \in D$: The set of vertices at step t, where each vertex v_i^t corresponds to a document d_i in corpus D.
- $E_t \subseteq V_t \times V_t$: The set of edges at step t connecting semantically similar document embeddings.

- $\omega_t : E_t \to \mathbb{R}$: A weight function assigning similarity scores to edges at step t.
- $\phi_t : V_t \to \mathbb{R}^d$: An embedding function at step t that assigns each vertex its current d-dimensional embedding vector.

As shown in Figure 4, our framework initializes the graph once with Stage 1 model embeddings, then selectively updates regions most affected by evolving model parameters during Stage 2 training. This allows the search environment to co-evolve with the model's embedding capability without the overhead of complete reconstruction. A naive approach would require such reconstruction after each parameter update, a process that is computationally infeasible at scale.

Algorithm 1 details our method for efficiently processing updates to topologically related embedding regions using a "local join" primitive. We first identify a critical subspace through two sequential nearest-neighbor searches: one targeting the positive example region and another focusing on the hard negative examples (lines 3-7). This dual-focused strategy ensures the capture of both the target retrieval neighborhood and the critical disambiguation regions that separate positives from hard negatives. These neighborhoods are further expanded to their 2-hop connections to utilize the transitivity of semantic similarity (lines 9-11). The final local join operation is efficient to apply graph changes, instead of sequentially updating individual nodes, this operation processes the entire neighborhood in a batch, simultaneously updating node embeddings and their connections.

Through this specialized environment design, our framework efficiently handles the continuously evolving embedding representations inherent in RL-based optimization.

6 Evaluation

6.1 Implementation

We train Search-R3 from Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct models, creating two variants: Search-R3-Small and Search-R3-Large, respectively. The post-training process employs Low-Rank Adaptation (LoRA) with r=32 to enable efficient parameter updates while maintaining model quality. We use AdamW optimizer with $\beta_1=0.9,\,\beta_2=0.999,$ weight decay of 0.01, and gradient clipping at 1.0. Training proceeds in two stages with tailored learning rates: 1e-5 for the supervised training phase (Stage 1) and 1e-6 for the RL phase (Stage 2). The Stage 1 training runs for 16384 total steps with a batch size of 32 sequences, each with maximum length of 2048 tokens, while Stage 2 performs 8192 rollout steps with group size of 16. We maintain bfloat16 precision during training for both the base model parameters and LoRA adaptation layers.

For training data, we curate a diverse mixture of sources as shown in Table 1. This mixture ensures broad domain coverage without potential contamination of evaluation benchmarks. The weight for each dataset is calculated as $\log(1.2 + \frac{Size}{1024})$, which ensures larger datasets receive proportionally more samples while preventing them from completely dominating the training mixture. Note that we include a Synthetic-100k dataset in our training, generated using Qwen3-32B to create hard-negative triplets, providing

Table 1: Training data composition.

Dataset	Size (Compressed, MiB)	Weight
TriviaQA [27]	30.4	0.21
Synthetic-100k	59.5	0.23
MSMARCO [8]	73.5	0.24
CodeSearchNet [25]	294.0	0.40
Miracl [76]	1035.9	0.79
S2ORC [39]	10829.3	2.47

high-quality training data derived from the other datasets as the datasource.

In total, training the Search-R3-Small requires approximately 105 GPU hours on RTX 4090 GPUs, while the training of the Search-R3-Large remains incomplete as we publish this manuscript. This represents a significantly smaller compute budget compared to commercial models, highlighting the training efficiency of Search-R3.

6.2 Experimental Setup

We evaluate Search-R3 on standard embedding retrieval benchmarks [45], focusing primarily on retrieval tasks. Our evaluation encompasses strong baseline models, and appropriate metrics to enable thorough performance analysis.

Baselines. We present our evaluation results by distinguishing between fully open-source models (with transparent training methodologies and data) and commercial/proprietary models (where aspects of training remain undisclosed). This separation addresses potential concerns regarding training data contamination and enables more meaningful comparisons between models developed under similar constraints. For open-source comparisons, we include BGE-M3 [7], Instructor [62], Sentence-T5 [46], and GTR-T5 [47].

Benchmarks. For retrieval evaluation, we use a diverse set of benchmarks to demonstrate Search-R3's generalization capabilities across different domains. CoSQA [23] evaluates code search performance, measuring the model's ability to match natural language queries with relevant code snippets. LitSearch [1] focuses on scientific literature search, MedicalQA [2] tests domain-specific retrieval in medical information, and SciFact [68] evaluates scientific claim verification through retrieving supporting or refuting documents. We also include MKQA [41], a challenging benchmark for assessing general question-answering capabilities.

Metrics. In the evaluation across all retrieval benchmarks, we utilize the asymmetric generation approach described in Section §5, where documents are encoded with a fixed template while queries undergo autoregressive processing. For retrieval quality, we report nDCG@k (k=1,10,100) and Recall@k (k=1,10,100). We highlight nDCG@10 as our primary metric, consistent with the MTEB benchmark's standard for retrieval tasks.

6.3 Main Results

Our evaluation demonstrates that Search-R3 achieves state-of-theart performance across diverse retrieval tasks, outperforming both leading open-source and proprietary embedding models.

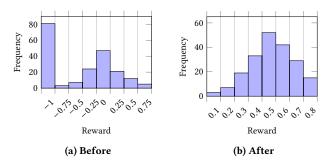


Figure 5: Score distributions before and after RL training.

Open-source models. Table 2 presents Search-R3's performance against prominent open-source embedding models on public benchmarks. Across 5 tasks, Search-R3 with reasoning enabled (indicated by green dots in the table) achieves substantial gains over baseline models. Specifically, Search-R3 improves the nDCG@10 from 0.194 to 0.211 on the most challenging MKQA evaluation. Interestingly, when reasoning is disabled, our model shows comparable performance to alternatives, with each model demonstrating particular strengths in specific domains. However, once reasoning is enabled, Search-R3 consistently outperforms all alternatives across benchmarks. This dramatic improvement with reasoning enabled is particularly evident in domain-specific tasks such as literature search (LitSearch, +0.036) and scientific claim retrieval (SciFact, +0.048). These results demonstrate that our approach to integrating reasoning capabilities into embedding representation provides a substantial advantage in semantic understanding.

Proprietary models. To ensure fair comparison with proprietary models where training data contamination concerns exist, we constructed a synthetic evaluation dataset consisting of 1,000 queries and 100,000 documents derived from Wikipedia [10, 24]. This dataset includes both positive matches and challenging negative examples that share topical similarity with relevant documents. Using Wikipedia as the source ensures the content falls within the knowledge domain of all evaluated models, while the synthetic curpus guarantees these exact sentences never appeared on the web, eliminating contamination concerns. As shown in Table 3, we observe a similar pattern: with reasoning disabled, Search-R3 performs comparably to proprietary alternatives, achieving an nDCG@10 of 0.858. However, when reasoning is enabled, our model achieves 0.871 nDCG@10, reaching Qwen3-Embedding-4B, which is $2 \times large than our model$.

6.4 Ablation Study

Effectiveness of RL. Figure 5 illustrates the impact of reinforcement learning on model performance. Before RL training, the model generates outputs with scores spanning -1.0 to 0.75, exhibiting a flatter distribution with an average score of -0.39. This substantial variance reflects the model has difficulty in stably generating high-quality reasoning paths and embedding tokens. After RL training, scores concentrate sharply around 0.5, with 69% of outputs achieving scores above 0.5. This transformation demonstrates that RL successfully guides the model toward consistently higher-quality outputs, resulting in more deterministic and reliable reasoning and

Table 2: Retrieval results of open-source models (green dot means reasoning is enabled).

Evaluation	Model	nDCG@1	nDCG@10 (mteb)	nDCG@100	Recall@1	Recall@10	Recall@100
	BGE-M3	0.100	0.267	0.348	0.100	0.494	0.874
CoSQA [23]	Instructor-XL	0.102	0.300	0.369	0.102	0.576	0.898
	Sentence-T5-XL	0.132	0.314	0.385	0.132	0.560	0.886
	GTR-T5-XL	0.116	0.298	0.369	0.116	0.549	0.872
	Search-R3-Small	0.124	0.315	0.384	0.124	0.550	0.904
	 Search-R3-Small 	0.124	0.317	0.387	0.132	0.549	0.901
	BGE-M3	0.297	0.427	0.471	0.294	0.577	0.789
	Instructor-XL	0.319	0.444	0.487	0.316	0.578	0.788
LitSearch [1]	Sentence-T5-XL	0.235	0.355	0.403	0.234	0.492	0.725
Litsearch [1]	GTR-T5-XL	0.245	0.354	0.397	0.243	0.477	0.689
	Search-R3-Small	0.303	0.417	0.464	0.302	0.547	0.765
	Search-R3-Small	0.326	0.453	0.496	0.323	0.590	0.793
	BGE-M3	0.526	0.680	0.705	0.526	0.830	0.944
	Instructor-XL	0.553	0.701	0.723	0.553	0.847	0.949
MedicalQA [2]	Sentence-T5-XL	0.436	0.608	0.640	0.436	0.787	0.936
MedicalQA [2]	GTR-T5-XL	0.528	0.692	0.713	0.528	0.851	0.949
	Search-R3-Small	0.543	0.714	0.732	0.543	0.882	0.967
	 Search-R3-Small 	0.546	0.716	0.734	0.546	0.885	0.971
	BGE-M3	0.042	0.068	0.099	0.125	0.102	0.245
	Instructor-XL	0.125	0.194	0.252	0.097	0.263	0.546
MKQA-eng [41]	Sentence-T5-XL	0.126	0.204	0.260	0.099	0.296	0.552
MKQA-eng [41]	GTR-T5-XL	0.115	0.181	0.240	0.083	0.265	0.538
	Search-R3-Small	0.127	0.189	0.227	0.099	0.261	0.433
	Search-R3-Small	0.151	0.211	0.255	0.118	0.285	0.481
	BGE-M3	0.510	0.644	0.672	0.482	0.783	0.907
	Instructor-XL	0.520	0.645	0.672	0.491	0.785	0.903
SciFact [68]	Sentence-T5-XL	0.397	0.509	0.557	0.370	0.648	0.874
SCIFACI [08]	GTR-T5-XL	0.537	0.642	0.680	0.510	0.748	0.921
	Search-R3-Small	0.503	0.624	0.657	0.478	0.761	0.913
	Search-R3-Small	0.560	0.672	0.704	0.535	0.795	0.933

Table 3: Retrieval results of proprietary models.

Model	nDCG@10
BGE-M3	0.843
GraniteEmbedding-278M [3]	0.842
EmbeddingGemma-300M [67]	0.723
Qwen3-Embedding-0.6B [77]	0.864
Qwen3-Embedding-4B [77]	0.879
Qwen3-Embedding-8B [77]	0.892
Search-R3-Small	0.858
Search-R3-Small	0.871

embedding representation.

Case Study on MSMARCO. We report one scenario we found Search-R3 demonstrates divergence from the MSMARCO dataset labels, i.e., after enabling reasoning, the retrieval performance degraded. As illustrated in Table 4, the query "which health care system provides all citizens or residents with equal access to health care services," the ground truth passage directly answers with "Universal Health Care" and provides its definition and scope. However, our model assigns a higher similarity score to a passage marked as negative, which discusses specific implementations in Singapore,

Table 4: Case study on MSMARCO.

Query	which health care system provides all citizens or residents with equal access to health care services?
Groundtruth	Universal Health Care which is also known as universal care, universal coverage or universal health coverage is a term that is used to address a health care system which provides health care and financial protection to every citizen of a specific country.
Prediction	In Singapore all residents receive a catastrophic policy from the government coupled with a health savings account that they use to pay for routine care. In other countries like Ireland and Israel, the government provides a core policy which the majority of the population supplement with private insurance.

Ireland, and Israel. Our model tends to prioritize passages that directly answer queries with explicit, concrete examples. In this case, during reasoning, specific keywords like "Singapore" are generated and contribute to the embedding representation, as the model recognizes these as essential context to healthcare system queries. While the ground truth passage delivers a concise definitional answer, our model recognizes the negative passage as more relevant due to its wider contextual coverage. This observation does not necessarily

indicate a quality problem with either the model or the dataset. Rather, it highlights that this type of search query often requires combining additional contextual signals such as user geographic location or search intent, and typically benefits from a reranking model after the initial retrieval stage.

7 Related Work

7.1 Language Model Adaptation

Adapting pre-ptrained LLMs to specialized tasks has produced several methodological paradigms, each offering distinct advantages. Parameter-efficient fine-tuning (PEFT) techniques, such as prefix tuning [31] and LoRA [22], adjust only a small fraction of model parameters while retaining the model's core functionality. A particularly relevant approach to our work is instruction tuning [29, 36, 58], which aligns models with tailored tasks or human preferences by training them on specialized examples. This method has shown substantial improvements in zero-shot generalization, as evidenced by models like FLAN [40] and Alpaca [64]. Additionally, SGPT [44], INSTRUCTOR [62], Instruction Embedding [32] and Qwen3-Embedding [77] explore instruction-tuned embedding representation but simply extract embeddings from model outputs without utilizing the sophisticated reasoning capabilities of LLMs. RankGPT [63] is a re-ranking model for retrieval tasks, this reranking model is different to our embedding models as it takes both the query and multiple candidates as input, and then directly produces a ranking score. Our method extends this paradigm and demonstrates that instruction tuning can teach LLMs to generate high-quality embeddings through their native token generation process.

7.2 Augmenting Language Models

LLMs are known to suffer from fundamental limitations including hallucination, outdated knowledge, and non-transparent reasoning processes, which significantly impact their reliability in knowledge-intensive applications. To address these challenges, Retrieval-Augmented Generation (RAG) has emerged as a prominent solution that incorporates knowledge from vast and dynamic external databases to enhance the accuracy and credibility of generation [16, 75, 79]. Existing RAG methods include iterative retrieval that enables multiple retrieval cycles for complex queries, and recursive retrieval that recursively iterates on outputs to process specific task demands [16]. Other approaches focus on memory enhancement through complementary frameworks such as LongMem [71], Camelot [20] and Larimar [11], which enable LLMs to memorize long history using tailored memory modules. Current research also focuses on Graph-based RAG [14, 19] that utilize structured knowledge representation to capture entity relationships and enable multihop reasoning through structure-aware knowledge retrieval. Our proposed Search-R3 is orthogonal to all these existing LLM augmentation methods, as it focuses on improving the fundamental representation mechanisms, and integration of our approach with these techniques can bring better system performance through enhanced semantic understanding and more effective model utilization.

8 Conclusion

This paper introduces a novel approach namely Search-R3 for transforming LLMs into powerful embedding generators. Our two-stage approach combines instruction-guided representation learning with reinforcement learning optimization, preserving the model's original architecture and functionality while enabling embedding generation capabilities. This allows AI-driven applications to use a single unified model for both generative and representative tasks, improving representation quality, reducing computational overhead and simplifying deployment. the unification of generative and representation learning capabilities advances the development of more versatile and efficient AI systems.

Acknowledgments

We acknowledge the use of AI (specifically Claude-Sonnet-3.7) solely for language editing and improving the clarity of presentation in this manuscript. All research ideas, methodology design, implementation, experiments, and analysis were conducted entirely by the authors without AI assistance. The authors take full responsibility for the content and integrity of this work.

References

- [1] Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. LitSearch: A Retrieval Benchmark for Scientific Literature Search. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 15068–15083. doi:10.18653/V1/2024.EMNLP-MAIN.840
- [2] Ben Abacha Asma and Demner-Fushman Dina. 2019. A Question-Entailment Approach to Question Answering. BMC Bioinform. 20, 1 (2019), 511:1–511:23. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4
- [3] Parul Awasthy, Aashka Trivedi, Yulong Li, Mihaela Bornea, David Cox, Abraham Daniels, Martin Franz, Gabe Goodhart, Bhavani Iyer, Vishwajeet Kumar, et al. 2025. Granite Embedding Models. arXiv preprint arXiv:2502.20204 (2025).
- [4] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016, Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith (Eds.). BMVA Press. https://bmva-archive.org.uk/bmvc/ 2016/papers/paper119/index.html
- [5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics* 5 (2017), 135–146. doi:10.1162/TACL_A_00051
- [6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
- [7] 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. CoRR abs/2402.03216 (2024). doi:10.48550/ARXIV.2402.03216 arXiv:2402.03216
- [8] Qi Chen, Xiubo Geng, Corby Rosset, Carolyn Buractaon, Jingwen Lu, Tao Shen, Kun Zhou, Chenyan Xiong, Yeyun Gong, Paul N. Bennett, Nick Craswell, Xing Xie, Fan Yang, Bryan Tower, Nikhil Rao, Anlei Dong, Wenqi Jiang, Zheng Liu, Mingqin Li, Chuanjie Liu, Zengzhong Li, Rangan Majumder, Jennifer Neville, Andy Oakley, Knut Magne Risvik, Harsha Vardhan Simhadri, Manik Varma, Yujing Wang, Linjun Yang, Mao Yang, and Ce Zhang. 2024. MS MARCO Web Search: A Large-scale Information-rich Web Dataset with Millions of Real Click Labels. In Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024, Tat-Seng Chua, Chong-Wah Ngo,

- Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw (Eds.). ACM, 292–301. doi:10. 1145/3589335.3648327
- [9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling Instruction-Finetuned Language Models. J. Mach. Learn. Res. 25 (2024), 70:1–70:53. https://jmlr.org/papers/v25/23-0870.html
- [10] Wikipedia contributors. 2025. Wikipedia, the free encyclopedia. https://en. wikipedia.org/wiki/Main_Page Accessed: 2025-10-01.
- [11] Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarath Swaminathan, Sihui Dai, Aurélie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Soham Dan, et al. 2024. Larimar: Large language models with episodic memory control. arXiv preprint arXiv:2403.11901 (2024).
- [12] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. CoRR abs/2501.12948 (2025). doi:10.48550/ARXIV.2501.12948 arXiv:2501.12948
- [13] 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. doi:10.18653/V1/N19-1423
- [14] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024).
- [15] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event Punta Cana, Dominican Republic, 7-11 November, 2021, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 6894–6910. doi:10.18653/V1/2021.EMNLP-MAIN.552
- [16] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 2, 1 (2023).
- [17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024).
- [18] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. arXiv preprint arXiv:2306.08543 (2023).
- [19] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-augmented generation with graphs (graphrag). arXiv preprint arXiv:2501.00309 (2024).
- [20] Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. 2024. Camelot: Towards large language models with training-free consolidated associative memory. arXiv preprint arXiv:2402.13449 (2024).
- [21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015).
- [22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9
- [23] Junjie Huang, Duyu Tang, Linjun Shou, Ming Gong, Ke Xu, Daxin Jiang, Ming

- Zhou, and Nan Duan. 2021. CoSQA: 20, 000+ Web Queries for Code Search and Question Answering. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 5690–5700. doi:10.18653/V1/2021.ACL-LONG.442
- [24] HuggingFaceFW. 2025. clean-wikipedia dataset at Hugging Face. https: //huggingface.co/datasets/HuggingFaceFW/clean-wikipedia Accessed: 2025-10-01.
- [25] Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. arXiv preprint arXiv:1909.09436 (2019).
- [26] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. CoRR abs/2503.09516 (2025). doi:10.48550/ARXIV. 2503.09516 arXiv:2503.09516
- [27] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 1601–1611. doi:10.18653/V1/P17-1147
- [28] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 6769–6781. doi:10.18653/V1/2020.EMNLP-MAIN.550
- [29] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant Conversations Democratizing Large Language Model Alignment. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 16, 2023, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/949f0f8f32267d297c2d4e3ee10a2e7e-Abstract-Datasets_and_Benchmarks.html
- [30] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net. https://openreview.net/forum?id=H1eA7AEtvS
- [31] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597. doi:10.18653/V1/2021.ACL-LONG.353
- [32] Yiwei Li, Jiayi Shi, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024. Instruction Embedding: Latent Representations of Instructions Towards Task Identification. In Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 15, 2024, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/9f9e2982759f384495f4b75a33f3dd72-Abstract-Datasets_and_Benchmarks_Track.html
- [33] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence 40, 12 (2017), 2935–2947.
- [34] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. CoRR abs/2308.03281 (2023). doi:10.48550/ARXIV.2308.03281 arXiv:2308.03281
- [35] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In The Twelfth International Conference on Learning Representations.
- [36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. Advances in neural information processing systems 36 (2023), 34892–34916.
- [37] Jerry Liu. 2022. *LlamaIndex*. doi:10.5281/zenodo.1234
- [38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR abs/1907.11692 (2019).

- arXiv:1907.11692 http://arxiv.org/abs/1907.11692
- [39] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 4969–4983. doi:10.18653/V1/2020.ACL-MAIN.447
- [40] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202), Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 22631–22648. https://proceedings.mlr.press/v202/longpre23a.html
- [41] Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering. https://arxiv.org/pdf/2007.15207.pdf
- [42] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html
- [43] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2022. Deep Learning-based Text Classification: A Comprehensive Review. ACM Comput. Surv. 54, 3 (2022), 62:1–62:40. doi:10.1145/3439726
- [44] N Muennighoff. [n. d.]. Sgpt: Gpt sentence embeddings for semantic search. arXiv 2022. arXiv preprint arXiv:2202.08904 ([n. d.]).
- [45] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. arXiv preprint arXiv:2210.07316 (2022). doi:10.48550/ARXIV.2210.07316
- [46] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 1864–1874. doi:10.18653/V1/2022.FINDINGS-ACL.146
- [47] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large Dual Encoders Are Generalizable Retrievers. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 9844–9855. doi:10.18653/V1/2022.EMNLP-MAIN.669
- [48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018).
- [49] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/ b1efde53be364a73914f58805a001731-Abstract-Conference.html
- [50] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. doi:10.3115/V1/D14-1162
- [51] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained Models for Natural Language Processing: A Survey. CoRR abs/2003.08271 (2020). arXiv:2003.08271 https://arxiv.org/abs/2003.08271
- [52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139), Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763. http://proceedings.mlr.press/v139/radford21a.html
- [53] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your

- Language Model is Secretly a Reward Model. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 16, 2023, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html
- [54] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. doi:10.18653/V1/D19-1410
- [55] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends® in Information Retrieval 3, 4 (2009), 333–389.
- [56] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. Trans. Assoc. Comput. Linguistics 8 (2020), 842–866. doi:10.1162/TACL_A_00349
- [57] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.
- [58] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207 (2021).
- [59] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. CoRR abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347
- [60] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. CoRR abs/2402.03300 (2024). doi:10.48550/ARXIV.2402.03300 arXiv:2402.03300
- [61] Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 60, 5 (2004), 493–502.
- [62] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 1102–1121. doi:10.18653/V1/2023.FINDINGS-ACL.71
- [63] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. arXiv preprint arXiv:2304.09542 (2023)
- [64] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html 3, 6 (2023), 7.
- [65] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. CoRR abs/2302.13971 (2023). doi:10.48550/ARXIV.2302.13971 arXiv:2302.13971
- [66] 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 (2017).
- [67] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, et al. 2025. EmbeddingGemma: Powerful and Lightweight Text Representations. arXiv preprint arXiv:2509.20354 (2025).
- [68] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 7534–7550. doi:10.18653/v1/2020.emnlp-main.609
- [69] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. CoRR abs/2212.03533 (2022). doi:10.48550/ARXIV.2212. 03533 arXiv:2212.03533
- [70] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 9426–9439. doi:10.18653/V1/2024.ACL-LONG.510
- [71] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory. Advances

- $in\ Neural\ Information\ Processing\ Systems\ 36\ (2023),\ 74530-74543.$
- [72] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 December 9, 2022, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html
- [73] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025).
- [74] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 Technical Report. CoRR abs/2412.15115 (2024). doi:10.48550/ARXIV.2412.15115 arXiv:2412.15115
- [75] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In CCF Conference on Big Data. Springer, 102–120.
- [76] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a miracl: Multilingual information retrieval across a continuum of languages. arXiv preprint arXiv:2210.09984 (2022).
- [77] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. CoRR abs/2506.05176 (2025). doi:10.48550/ARXIV.2506.05176 arXiv:2506.05176
- [78] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International* Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net. https://openreview.net/forum?id=5NTt8GFjUHkr
- [79] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrievalaugmented generation for ai-generated content: A survey. arXiv preprint arXiv:2402.19473 (2024).