Semi-Parametric Retrieval via Binary Bag-of-Token Index

Jiawei Zhou, Li Dong, Furu Wei, Lei Chen

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Microsoft Research

TL;DR:

Semi-parametric Disentangled Retrieval (SiDR) [1] is a novel bi-encoder retrieval architecture designed for emerging RAG applications, addressing the growing need for:

(i) **efficient indexing [2]** to enable **RAG retrieve from real-time data sources**.

(ii) **low-cost indexing [3]** to support **resource-constrained RAG deployments and exploration**.

(iii) **non-parametric index** to facilitate **co-training with LLMs [4]**.

[4] Find our follow-up paper in workshop {Data-FM, SSI-FM, FM-Wild}
"Optimizing RAG End-to-end via In-context Retrieval Learning"

[1] Methodology

Background: SiDR represents data in a vocabulary space — a 30,522-dimensional vector (BERT's vocab size) where each dimension corresponds to the importance weight of a specific token.

Key idea: SiDR employs two representation methods
1. Embedding representation *E(x)*, same as neural retriever.
2. Bag-of-tokens representation *T(x)*, which only uses tokenizer.

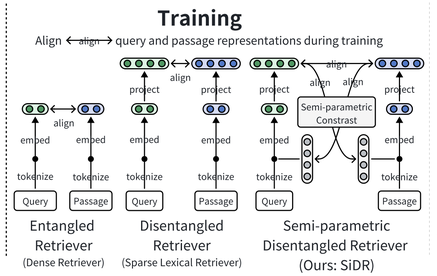
E(x) can be viewed as a 30522-dimensional representation with learned term weighting, while T(x) can be viewed as unweighted.

During Training:

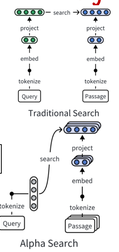
SiDR uses two representations:
- E(·): Embedding-based (semantic)
- T(·): Tokenization-based (lexical)

Contrastive Learning Objectives:
Align queries (q) and documents (d) across:
1. E(q) ⇔ E(d) # embedding match
2. E(q) ⇔ T(d) # support tokenization-based index
3. T(q) ⇔ E(d) # support tokenization-based query

Downstream Benefit:
Enables flexible search pipeline for diverse search or RAG scenarios.



During Inference: SiDR support multiple search pipelines:



i. Traditional Search: embedded query E(q) search embedded documents E(d)

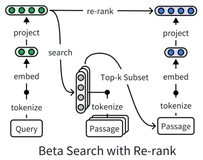
Similar to existing sparse lexical retrievers, it requires embedding both queries and documents prior to search.

ii. Alpha Search: tokenized query T(q) search embedded documents E(d)

No need to embed queries!
This approach is suitable when the data is indexed and there is a need to reduce the online query processing workload.

iii. Beta Search: embedded query E(q) search tokenized documents T(d)

No need to embed document collection!
This approach is ideal when dealing with large-scale or rapidly changing datastores (e.g., real-time updates or exploratory RAG applications).

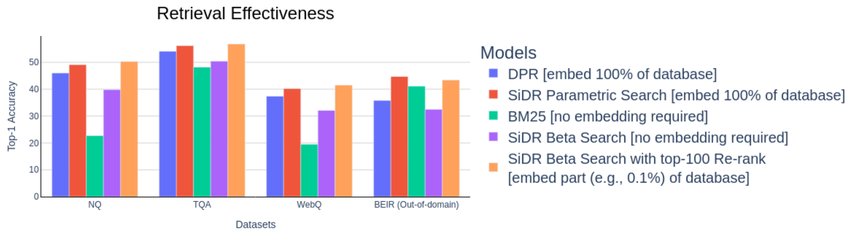


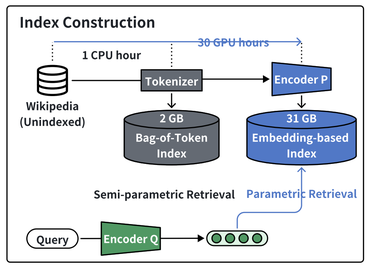
iv. Beta Search with Re-rank (late-parametric):
1. E(q) search top-k documents from T(d)
2. E(q) re-rank with top-k documents embedding E(d)

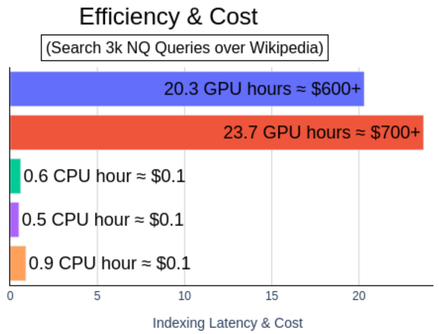
Embed a small portion of document, gain significant improvement!

With only 0.00001x indexing workload, this approach achieves search accuracy comparable to traditional neural retriever

[2,3] Experimental Results (Effectiveness/Efficiency/Cost)





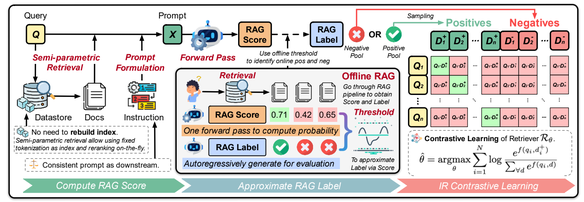


[4] Co-training with LLMs

For further details, please refer to our arXiv preprint or the upcoming workshop paper:

OPEN-RAG: Optimizing RAG End-to-End via In-Context Retrieval Learning

To appear in the following workshops:
- SSI-FM: Scaling Self-Improving Foundation Models without Human Supervision
- FM-Wild: Foundation Models in the Wild
- Data-FM: Navigating and Addressing Data Problems for Foundation Models



When using embedding-based index:
SiDR significantly outperforms vanilla neural retriever with the same parameters and size.

When using tokenization-based index:
SiDR achieves significant performance improvements over BM25.

Late parametric = tokenization-based index + embedding-based re-ranking
By investing minimal computation to re-rank the top-100 retrieved passages, SiDR outperforms retrieval methods of comparable complexity with ***less than 0.1% of the indexing cost***.