Taxonomy-guided Semantic Indexing for Academic Paper Search

SeongKu Kang[1], **Yunyi Zhang**[1], **Pengcheng Jiang**[1], **Dongha Lee**[2],
Jiawei Han[1], **Hwanjo Yu**[3*]
[1]University of Illinois at Urbana Champaign
[2]Yonsei University [3]Pohang University of Science and Technology
{seongku,yzhan238,pj20,hanj}@illinois.edu
donalee@yonsei.ac.kr hwanjoyu@postech.ac.kr

Abstract

Academic paper search is an essential task for efficient literature discovery and scientific advancement. While dense retrieval has advanced various ad-hoc searches, it often struggles to match the underlying academic concepts between queries and documents, which is critical for paper search. To enable effective academic concept matching for paper search, we propose **Taxo**nomy-guided semantic **Index**ing (TaxoIndex) framework. TaxoIndex extracts key concepts from papers and organizes them as a *semantic index* guided by an academic taxonomy, and then leverages this index as foundational knowledge to identify academic concepts and link queries and documents. As a plug-and-play framework, TaxoIndex can be flexibly employed to enhance existing dense retrievers. Extensive experiments show that TaxoIndex brings significant improvements, even with highly limited training data, and greatly enhances interpretability.

1 Introduction

Academic paper search is essential for efficient literature discovery and access to technical solutions. Recently, dense retrieval has advanced in various ad-hoc searches (Karpukhin et al., 2020; Izacard et al., 2021). It encodes queries and documents as dense embeddings, measuring relevance by embedding similarity. These embeddings effectively capture textual meanings via pre-trained language models trained on massive corpora. While effective in general domains like web search, it often shows limitations in paper search (Wang et al., 2023).

In paper search, it is crucial to match the underlying academic concepts between queries and documents, rather than relying on surface text and its meanings. Academic concepts refer to fundamental ideas, theories, and methodologies that constitute the contents of papers. Users often seek information on specific concepts when searching for papers. For example, consider the query "*learning to win by reading manuals in a Monte-Carlo framework*". This query encompasses various concepts: optimizing decision-making (learning to win), acquiring knowledge from text (reading manuals), and reinforcement learning using probabilistic sampling (Monte-Carlo). Accordingly, retrievers should find papers that comprehensively cover these concepts.

One critical limitation of existing dense retrievers is that such academic concepts are often not effectively captured, making them insufficiently considered in relevance prediction. Identifying underlying concepts from surface text requires an inherent understanding of domain-specific contents, which is not sufficiently obtained from general corpora. This challenge is even greater for queries. As shown in the previous example, user queries often encompass various academic concepts in highly limited contexts. Moreover, queries usually have different expression styles (e.g., terminology choice, language style) from documents, making it difficult to match common concepts.

Figure 1(left) shows the results of a dense retriever for the query, which ranks an irrelevant document (Paper A) at the top-1. Although the paper mentions 'solving a goal' and 'comprehension of text', which have similar meanings to 'learning to win' and 'reading manuals' in the query, its focus is on text comprehension and dataset creation, which largely differs from the query concepts. This shows that the overall textual similarity captured by dense retrievers is insufficient to reflect specific academic concepts, leading to suboptimal results.

To address this limitation, we introduce a new approach that extracts key concepts from papers in advance, and leverages this knowledge to incorporate academic concepts into relevance predictions. We construct a *semantic index* that stores semantic components best describing each paper. The pro-

*Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7169–7184
November 12-16, 2024

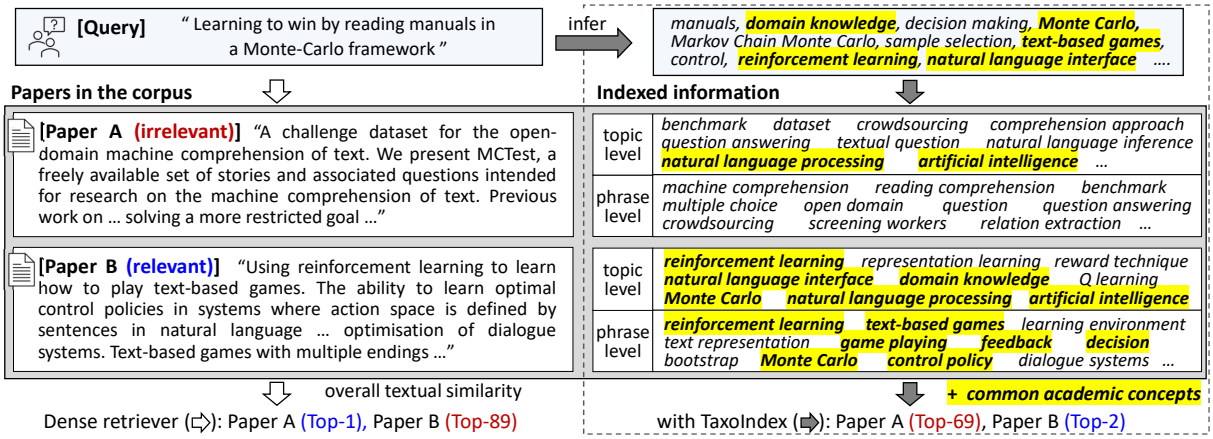


Figure 1: A case study from CSFCube dataset. Results of (left) a dense retriever, (right) with TaxoIndex. For the dense retriever, we use SPECTER-v2 fully fine-tuned on the target corpus.

posed index represents each paper at two different granularities: topic and phrase levels, as shown in Figure 1(right). Topic level provides a broader categorization of research, such as 'reinforcement learning' or 'natural language processing', while phrase level includes specific details, such as 'text-based games' or 'control policy', complementarily revealing each paper's concepts. We leverage this index to enhance the existing dense retrievers, enabling more precise academic concept matching.

We propose **Taxo**nomy-guided semantic **Index**ing (TaxoIndex) framework. We first introduce a new index construction strategy that extracts key concepts from papers. To guide this process, we propose using an academic taxonomy, a hierarchical tree structure outlining academic topics.[1] We then propose a new training strategy, called *index learning*, that trains a model to explicitly identify the indexed information for input text. This is a critical technique that enables TaxoIndex to infer the most related academic concepts in test queries, even if expressed in different terms or not explicitly mentioned, by associating them with papers having similar contexts. This inferred information helps find relevant papers sharing academic concepts, combined with textual similarity from dense retrievers. Figure 1(right) shows that TaxoIndex finds Paper B as the top-2 result based on high overlap in indexed information.

TaxoIndex offers several advantages for existing dense retrievers. First, TaxoIndex largely improves retrieval quality by effectively matching academic concepts. Notably, index learning offers a new type of supervision to identify key concepts from text, which is not directly provided by query-document training pairs, yielding significant improvements even with limited training data. Second, TaxoIndex enhances interpretability by explicitly representing each text with topics and phrases, as shown in our case studies. Further, these advancements do not come at a large cost of model complexity. TaxoIndex updates only a small module, accounting for 6.7% of the retriever parameters, yet outperforms fully fine-tuned models.

Our primary contributions are as follows: (1) We propose TaxoIndex, which systematically constructs and leverages semantic index, for effective academic concept matching in paper search. (2) We design an index construction strategy that represents each paper at both topic and phrase levels, with the guidance of academic taxonomy. (3) We introduce an index learning strategy that allows for identifying the most related concepts from an input text. (4) We evaluate TaxoIndex with extensive quantitative and ablative experiments and comprehensive case studies.

2 Related Work

Dense retrieval. The advancement of pre-trained language models (PLMs) has led to significant progress in dense retrieval. Recent studies have enhanced retrieval quality through retrieval-oriented pre-training (Izacard et al., 2021; Gao and Callan, 2022), advanced hard negative mining (Zhan et al., 2021; Qu et al., 2021), and distillation from cross-encoder (Zhang et al., 2022). Synthetic query generation has also been explored to supplement training data (Thakur et al., 2021; Dai et al., 2023).

[1] Academic taxonomies are widely used for study categorization in various institutions (e.g., ACM Computing Classification System) and can be readily obtained from the web.

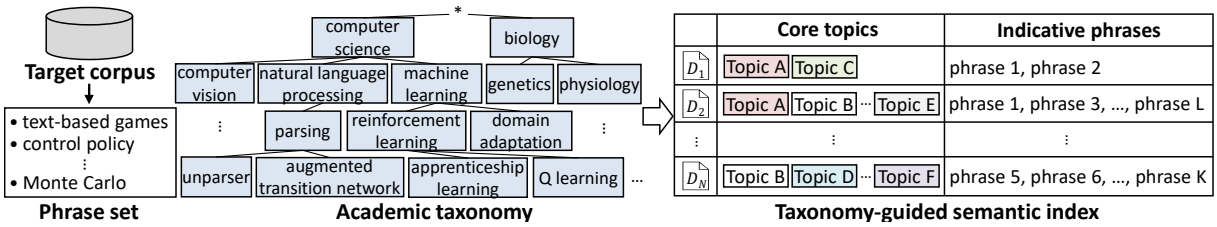


Figure 2: A conceptual illustration of the taxonomy-guided semantic index construction. We extract and store core topics and indicative phrases that best represent each paper in the form of a forward index.

On the other hand, many studies have focused on pre-training methods specialized for the academic domain. In addition to pre-training on academic corpora (Beltagy et al., 2019), researchers have exploited metadata associated with papers. Cohan et al. (2020); Ostendorff et al. (2022) use citations, Liu et al. (2022) further utilizes venues, authors, and affiliations. Mysore et al. (2022) uses co-citation contexts, and Singh et al. (2023); Zhang et al. (2023b) employs multi-task learning of tasks such as citation prediction and paper classification.

Complementary to the approach of leveraging such paper metadata, we focus on organizing and exploiting knowledge in the textual corpus. TaxoIndex can be flexibly integrated to enhance the aforementioned models.

Indexing for dense retrieval. Indexing refers to the process of collecting, parsing, and storing data to enhance retrieval (Moura and Cristo, 2009). Statistical and sparse retrieval often uses inverted indexes for term matching signals (Bruch et al., 2024). Dense retrieval relies on approximate nearest neighbor (ANN) indexes to avoid costly brute-force searches. Document embeddings are pre-computed offline, and ANN indexes are constructed by techniques such as hashing (Pham and Liu, 2022), quantization (Baranchuk et al., 2018), and clustering (Zhang et al., 2023a; Li et al., 2023).

Our index is designed to extract academic concepts and leverage them to improve the accuracy of dense retrievers. As TaxoIndex encodes each text as an embedding, existing ANN indexes can still be applied to accelerate search speed.

Enhancing retrieval with additional contexts. Several studies have enhanced retrieval by providing supplementary contexts. Our work falls into this direction. Pseudo-relevance feedback (Zheng et al., 2020; Wang et al., 2021; Yu et al., 2021) utilizes the top-ranked results from an initial retrieval. Recent generative approaches (Mao et al., 2021; Mackie et al., 2023) generate relevant contexts using PLMs. Kang et al. (2024) utilizes topic distributions of queries and documents. However, they are often limited in paper search due to the difficulty of generating proper domain-specific contexts. Moreover, these contexts are obtained and added on-the-fly during inference, making it difficult to provide information tailored to backbone retriever.

3 Problem Formulation

Academic taxonomy. An academic taxonomy $\mathcal{T}$ refers to a hierarchical tree structure outlining academic topics (Figure 2). Each node represents an academic topic, with child nodes corresponding to its sub-topics. Widely used for study categorization in various institutions, academic taxonomies can be readily obtained from the web and automatically expanded by identifying new topics from a growing corpus (Lee et al., 2022; Xu et al., 2023). We utilize the fields of study taxonomy from Microsoft Academic (Shen et al., 2018), which covers 19 disciplines (e.g., computer science, biology).

Problem definition. To perform retrieval on a new corpus $\mathcal{D}$, a PLM-based dense retriever is typically fine-tuned using a training set of relevant query-document pairs. Our goal is to develop a plug-and-play framework, which facilitates academic concept matching with the guidance of a given taxonomy $\mathcal{T}$, to improve the backbone retriever.

4 TaxoIndex Framework

We present taxonomy-guided index construction in §4.1, index-grounded fine-tuning in §4.2, and retrieval process with TaxoIndex in §4.3.

4.1 Taxonomy-guided Index Construction

We construct a semantic index that stores semantic components that best describe each paper (Figure 2). To guide this process, we propose using the academic taxonomy. This ensures that the index organizes knowledge in alignment with the researchers' consensus and greatly improves interpretability.

Our key idea is to represent each paper using a combination of *core topics* and *indicative phrases* that reveal its key concepts at different granularities. Core topics correspond to nodes in the taxonomy, providing a broader view for categorizing papers. Indicative phrases are directly extracted from each paper, offering finer-grained information to distinguish it from other topically similar documents.

4.1.1 Core Topic Identification

The given taxonomy may contain many topics not included in the corpus. To effectively identify core topics from the vast topic hierarchy, we introduce a two-step strategy that first finds candidate topics and then pinpoints the most relevant ones.

Candidate topics identification. Utilizing the hierarchy, we employ a top-down traversal approach that *recursively visits* the child nodes with the highest similarities at each level. For each document, we start from the root node and compute its similarity to each child node. We then visit child nodes with the highest similarities.[2] This process recurs until every path reaches leaf nodes, and *all visited nodes* are regarded as candidates for the document.

The document-topic similarity $s(d, c)$ can be defined in various ways. As a topic includes its subtopics, we incorporate the information from all subtopics for each topic node. Let $\mathcal{N}_c$ denote the set of nodes in the sub-tree having c as a root node. We compute the similarity as: $s(d, c) = \frac{1}{|\mathcal{N}_c|} \sum_{j \in \mathcal{N}_c} \cos(\mathbf{e}_d, \mathbf{e}_j)$, where $\mathbf{e}_d$ and $\mathbf{e}_j$ denote representations from PLM for a document d and the topic name of node j, respectively.[3]

Core topic selection. We select core topics by filtering out less relevant ones from the candidates. We consider two strategies: (1) score-based filtering, which retains topics with similarities above a certain threshold, and (2) LLM-based filtering, which uses large language models (LLMs) to select core topics. Our preliminary analysis shows that both filtering strategies are effective and lead to comparable retrieval accuracy. In this work, we opt for LLM-based filtering, as it often handles ambiguous cases better, further enhancing retrieval interpretability. Further analysis is provided in §5.3.

We prompt the LLM to select core topics from the candidates by excluding those that are too broad or less relevant.[4] After identifying core topics for all documents, we tailor the taxonomy by only retaining the topics selected as core topics at least once, along with their ancestor nodes.

In sum, for each document d, we obtain core topics as $\mathbf{y}_d^t \in \{0, 1\}^{|\mathcal{T}_\mathcal{D}|}$, where $y_{di}^t = 1$ indicates i is a core topic of d, otherwise 0. $|\mathcal{T}_\mathcal{D}|$ denotes the number of nodes in the tailored taxonomy.

4.1.2 Indicative Phrase Extraction

From each document, we extract indicative phrases used to describe its key concepts. These phrases offer fine-grained details not captured by topic level, playing a crucial role in understanding detailed content and enhancing retrieval. An indicative phrase should (1) show stronger relevance to the document than to others with similar core topics, and (2) refer to a meaningful and understandable notion.

We first obtain the phrase set $\mathcal{P}$ in the corpus using an off-the-shelf phrase mining tool (Shang et al., 2018). Then, inspired by Tao et al. (2016); Lee et al. (2022), we compute the indicativeness of phrase p in document d based on two criteria: (1) Distinctiveness $dist(p, \mathcal{D}_d) = \exp(\text{BM25}(p, d)) / (1 + \sum_{d' \in \mathcal{D}_d} \exp(\text{BM25}(p, d')))$ quantifies the relative relevance of p to the document d compared to other topically similar documents $\mathcal{D}_d$. $\mathcal{D}_d$ is simply retrieved using Jaccard similarity of core topic annotation $\mathbf{y}_d^t$. (2) Integrity $int(p)$ measures the conceptual completeness of the phrase, typically provided by most phrase mining tools, preventing the selection of non-meaningful phrases. The final indicativeness of p is defined as: $(dist(p, \mathcal{D}_d) \cdot int(p))^{\frac{1}{2}}$.

For each document d, we select top-k indicative phrases and denote them as $\mathbf{y}_d^p \in \{0, 1\}^{|\mathcal{P}|}$, where $y_{dj}^p = 1$ indicates j is an indicative phrase of d.

Remarks. Compared to recent clustering-based indexes for dense retrieval (Zhan et al., 2022; Li et al., 2023), which use cluster memberships from document clustering, the proposed index has several strengths: it effectively exploits domain knowledge from taxonomy, offers broad and detailed views via topics and phrases, and enhances interpretability.

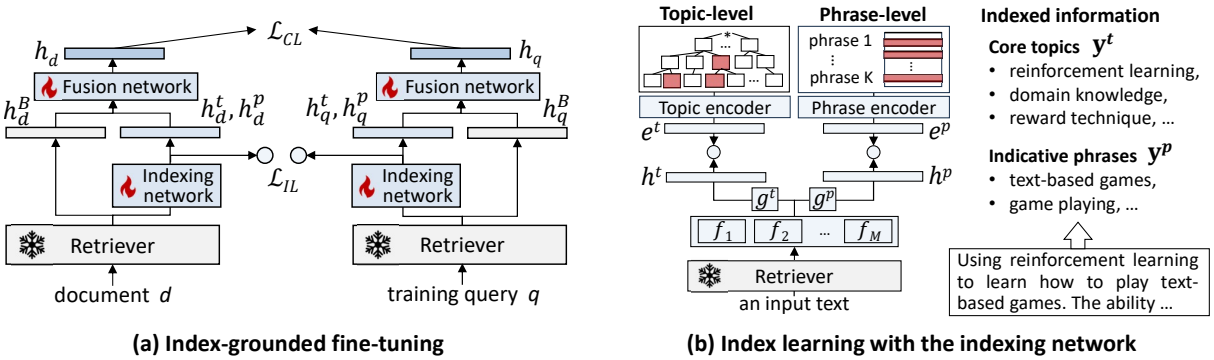
4.2 Index-grounded Fine-tuning

We train an add-on module to enhance relevance prediction while keeping the backbone retriever frozen (Figure 3). It comprises an indexing network and a fusion network, and is applied identically to

[2] We visit multiple child nodes and create multiple paths, as a document usually covers various topics. For a node at level l, we visit $l + 2$ nodes to reflect the increasing number of nodes at deeper levels of the taxonomy. The root node is level 0.

[3] We use BERT with mean pooling as the simplest choice.

[4] The prompt can be found in Appendix A.



(a) Index-grounded fine-tuning **(b) Index learning with the indexing network**

Figure 3: An illustration of TaxoIndex: (a) index-grounded fine-tuning, (b) index learning with the indexing network.

both documents and queries using shared parameters. Here, we describe it on the document side.

4.2.1 Indexing Network: linking text to index

A naive approach to using the index information is to append it to each text as additional input context. However, this approach has several limitations. Importantly, test queries are not accessible before the test phase. Annotating topics and phrases during inference not only incurs additional latency but also is less effective due to the limited context of queries.

As a solution, we propose a new strategy called index learning, which trains the indexing network to identify core topics and indicative phrases from the text. We formulate this as two-level classification tasks, i.e., topic and phrase levels.

Extracting topic/phrase information. Given the backbone retriever embedding $\mathbf{h}_d^B \in \mathbb{R}^l$, we extract information tailored to predict topics and phrases as $\mathbf{h}_d^t$ and $\mathbf{h}_d^p$, respectively. To exploit the complementarity of topics and phrases, we employ a multi-gate mixture of experts architecture (Ma et al., 2018).

We use M different experts, $\{f_m\}_{m=1}^M$, each of which is a small feed-forward network $f_m : \mathbb{R}^l \rightarrow \mathbb{R}^l$. Two gating networks, g^t and g^p, with Softmax outputs control the influence of experts for topic and phrase prediction, respectively. Let $\mathbf{w}^t = g^t(\mathbf{h}_d^B)$ and $\mathbf{w}^p = g^p(\mathbf{h}_d^B)$ denote M-dimensional vectors controlling the influences. The representations for each task are computed as:

$$\mathbf{h}_d^t = \sum_{m=1}^{M} w_m^t f_m(\mathbf{h}_d^B), \;\; \mathbf{h}_d^p = \sum_{m=1}^{M} w_m^p f_m(\mathbf{h}_d^B) \quad (1)$$

This enables the direct sharing of information beneficial for predicting both topics and phrases, mutually enhancing both tasks (Ma et al., 2018).

Generating class representation. We encode topics and phrases to generate class representations for classification learning. For topics, we employ graph neural networks (GNNs) (Kipf and Welling, 2016) to exploit their hierarchical information. Initially, each node feature is set as the fixed PLM representation of its topic name, followed by GNN propagation over the taxonomy structure. After stacking GNN layers, we obtain each topic representation $\mathbf{e}_i^t$. Each phrase is also encoded using PLM as $\mathbf{e}_j^p$. For notational simplicity, we pack the topic and phrase representations, denoting them as $\mathbf{E}^t \in \mathbb{R}^{|\mathcal{T}_\mathcal{D}| \times l}$ and $\mathbf{E}^p \in \mathbb{R}^{|\mathcal{P}| \times l}$, respectively.

Index learning. For each input text, index learning is applied to predict the corresponding core topics and indicative phrases $\mathbf{y}_d^t$ and $\mathbf{y}_d^p$. We compute the probabilities for topics and phrases as $\hat{\mathbf{y}}_d^t = \text{softmax}(\mathbf{E}^t(\mathbf{h}_d^t)^T)$, $\hat{\mathbf{y}}_d^p = \text{softmax}(\mathbf{E}^p(\mathbf{h}_d^p)^T)$, respectively. The cross-entropy loss is then applied:

$$\mathcal{L}_{IL} = -\sum_{i=1}^{|\mathcal{T}_\mathcal{D}|} y_{di}^t \log \hat{y}_{di}^t - \sum_{j=1}^{|\mathcal{P}|} y_{dj}^p \log \hat{y}_{dj}^p \quad (2)$$

The index learning can also be applied to training queries, with topic and phrase labels obtained in two ways: (1) through a separate annotation process, or (2) by using labels of the relevant documents, assuming that they reveal the details of the query. In this work, we choose the latter approach.

4.2.2 Fusion Network: fusing index knowledge

The index-based representations $(\mathbf{h}_d^t, \mathbf{h}_d^p)$ encode core topics and indicative phrases comprising the academic concepts within the text. We fuse them with the backbone embedding $(\mathbf{h}_d^B)$, which encodes the overall textual meanings, to generate $\mathbf{h}_d \in \mathbb{R}^l$.

We combine the topic and phrase representations as $\mathbf{h}_d^I = f^I([\mathbf{h}_d^t; \mathbf{h}_d^p])$ using a small network $f^I : \mathbb{R}^{2l} \rightarrow \mathbb{R}^l$. The final embedding is obtained as:

$$\mathbf{h}_d = \mathbf{h}_d^B + \alpha \cdot w_d \cdot \mathbf{h}_d^I \quad (3)$$

To effectively fuse the two information types, we devise a two-level trainable weight scheme: (1) a global weight α, and (2) an input-adaptive weight $w_d = \text{Sigmoid}(g^I(\mathbf{h}_d^B))$ using a small network g^I. During training, this technique emphasizes index information for input text where predicting relevance from backbone embedding is challenging.

4.2.3 Fine-tuning with TaxoIndex

We train the add-on module using the standard contrastive learning $\mathcal{L}_{CL}$ with our index learning $\mathcal{L}_{IL}$. For each query q, the contrastive learning loss is:

$$-\log \frac{e^{\text{sim}(\mathbf{h}_q, \mathbf{h}_{d^+})}}{e^{\text{sim}(\mathbf{h}_q, \mathbf{h}_{d^+})} + \sum_{d^-} e^{\text{sim}(\mathbf{h}_q, \mathbf{h}_{d^-})}} \quad (4)$$

where d^+ and d^- denote the relevant and irrelevant documents. Index learning is applied to both documents and training queries Q. The final objective is $\mathcal{L}_{CL}(Q, \mathcal{D}) + \lambda_{IL}(\mathcal{L}_{IL}(\mathcal{D}) + \mathcal{L}_{IL}(Q))$, where λ_{IL} is a hyperparameter to balance the loss. To ensure $\mathbf{h}_d^I$ contains high-quality information, we initially warm up the indexing network using $\mathcal{L}_{IL}$.

Core topic-aware negative mining. We devise a new strategy that uses core topics to mine hard-negative documents. Core topics reveal key concepts based on taxonomy, which may not be effectively captured by the lexical overlap (e.g., BM25) widely used for negative mining (Formal et al., 2022). We utilize both topical and lexical overlaps to select negative documents. For each (q, d^+) pair, we retrieve $\mathcal{D}_{d^+}$, a set of topically similar documents to d^+, using Jaccard similarity of core topics, as done in §4.1.2. We then select documents with the highest BM25 scores for q as negative samples.

4.3 Retrieval with TaxoIndex

Based on the index, TaxoIndex incorporates the similarity of surface texts and the similarity of the most related concepts for relevance prediction. This approach enhances the understanding of test queries, enables more precise academic concept matching, and improves paper search.

We introduce advanced inference techniques to further enhance retrieval using topic/phrase predictions $(\hat{\mathbf{y}}^t, \hat{\mathbf{y}}^p)$ for queries and documents.

Document filtering based on core topics. Before applying the retriever, we filter out irrelevant documents that have minimal core topic overlap with the query. This step enhances subsequent retrieval by reducing the search space and providing topical overlap information. We compute the topical overlap using the inner product of $\hat{\mathbf{y}}_q^t$ and $\hat{\mathbf{y}}_d^t$. Documents with low topical overlap are excluded, retaining only the top $x\%$ of documents from the entire corpus. In this work, we set $x = 25\%$. We provide retrieval results with varying x in §5.3.

Interpreting search results. The topics and phrases with the highest probabilities reveal the academic concepts captured and reflected in relevance prediction. Comparing query and document predictions allows for interpreting the search results. We provide case studies in Figure 1 and Appendix C.3.

Expanding query with indicative phrases. We can expand a query by appending top-k phrases not included in the query. The retrieval results using the expanded query are denoted as TaxoIndex ++.

5 Experiments

5.1 Experiment setup

We provide further details on setup in Appendix B.

Dataset and taxonomy. We use two datasets[5]: CSFCube (Mysore et al., 2021) and DORIS-MAE (Wang et al., 2023), which provide test query collections along with relevance labels on the academic corpus, annotated by human experts and LLMs, respectively. We use training queries generated by Dai et al. (2023), as they are not provided in both datasets. We use the field of study taxonomy from Microsoft Academic (Shen et al., 2018) which contains 431,416 nodes. After indexing, we obtain 1,164 topics and 3,966 phrases for CSFCube, and 1,498 topics and 6,851 phrases for DORIS-MAE. For core topic selection in TaxoIndex and baselines that require LLMs, we use `gpt-3.5-turbo-0125`.

Metrics. Following Mackie et al. (2023); Kang et al. (2024), we employ Recall@K (R@K) for a large retrieval size (K), and NDCG@K (N@K) and MAP@K (M@K) for a smaller K (≤ 10).

Backbone retrievers. We employ two representative models: (1) SPECTER-v2 (Singh et al., 2023) is a highly competitive model trained using metadata of scientific papers. (2) Contriever-MS (Izacard et al., 2021) is a widely used retriever fine-tuned using vast labeled data from general domains.

Baselines. We compare three types of methods for applying and improving the backbone retriever.

(1) Conventional approaches: **no Fine-Tuning**, **Full Fine-Tuning (FFT)**, **add-on module Fine-Tuning (aFT)**. FFT and aFT follow standard con-

[5]We provide results on SCIDOCS dataset in BEIR benchmark (Thakur et al., 2021) in Appendix C.1.

		CSFCube						DORIS-MAE					
		N@5	N@10	M@5	M@10	R@50	R@100	N@5	N@10	M@5	M@10	R@50	R@100
	BM25	0.307	0.310	0.088	0.134	0.504	0.635	0.354	0.330	0.079	0.107	0.490	0.669
SPECTER-v2	no Fine-Tuning	0.352	0.337	0.108	0.151	0.524	0.680	0.385	0.360	0.079	0.113	0.551	0.709
	FFT	0.372	0.368	0.123	0.169	0.576	0.692	0.408	0.387	0.084	0.122	0.562	0.736
	aFT	0.378	0.344	0.119	0.160	0.578	0.696	0.400	0.372	0.080	0.115	0.558	0.714
	FFT w/ GRF	0.331	0.317	0.112	0.152	0.561	0.705	0.400	0.379	0.087	0.123	**0.586**	**0.756**
	FFT w/ ToTER	0.406	0.375	0.135	0.179	0.591	0.710	0.423	0.394	0.091	0.128	0.563	0.736
	JTR	0.379	0.352	0.118	0.157	0.598	0.699	0.395	0.380	0.080	0.118	0.548	0.713
	TaxoIndex	0.458†*	0.417†*	0.144†*	0.198†*	**0.633**†*	0.741†*	0.447†*	0.421†*	0.104†*	0.144†*	0.578†	**0.756**†
	TaxoIndex ++	**0.469**†*	**0.426**†*	**0.158**†*	**0.209**†*	0.621†*	**0.746**†*	**0.449**†*	**0.424**†*	**0.105**†*	**0.145**†*	0.581†	0.751†
Contriever-MS	no Fine-Tuning	0.340	0.311	0.095	0.130	0.551	0.682	0.427	0.398	0.084	0.124	0.507	0.635
	FFT	0.364	0.328	0.110	0.149	0.589	0.705	0.453	0.407	0.093	0.132	0.510	0.652
	aFT	0.346	0.328	0.104	0.151	0.578	0.711	0.439	0.402	0.089	0.128	0.507	0.648
	FFT w/ GRF	0.353	0.313	0.108	0.136	0.559	0.669	0.447	0.381	0.090	0.120	0.511	0.643
	FFT w/ ToTER	0.375	0.353	0.121	0.169	**0.597**	0.724	0.458	**0.423**	0.097	0.139	0.539	0.703
	JTR	0.351	0.331	0.105	0.152	0.578	0.714	0.435	0.408	0.089	0.132	0.516	0.667
	TaxoIndex	0.400†*	**0.386**†*	0.135†*	0.184†*	0.596†	**0.726**†	0.461	**0.423**†	0.100	0.141†	0.557†*	0.729†*
	TaxoIndex ++	**0.421**†*	**0.386**†*	**0.144**†*	**0.185**†*	0.595	**0.726**†	**0.463**	0.422†	**0.101**	**0.142**†	**0.560**†*	**0.733**†*

Table 1: Retrieval performance comparison on CSFCube and DORIS-MAE datasets. † and * indicate the statistically significant difference (paired t-test, $p < 0.05$) from FFT and the best baseline, respectively.

trastive learning with BM25 negatives. FFT updates the entire backbone retriever, while aFT only updates an add-on module identical to TaxoIndex.[6]

(2) Enhancing retrieval with additional context: **GRF** (Mackie et al., 2023) generates relevant contexts by LLMs. We generate both topics and keywords for a fair comparison. **ToTER** (Kang et al., 2024) uses the similarity of topic distributions between queries and documents, with topics provided by the taxonomy. We apply both methods to FFT.

(3) Enhancing fine-tuning using an index: **JTR** (Li et al., 2023) constructs a tree-based index via clustering, then jointly optimizes the index and text encoder. Though focused on efficiency, it also enhances accuracy with index-based learning. We impose no latency constraints for a fair comparison.

We provide details on hyperparameters and implementation in Appendix B.4.

5.2 Retrieval Performance Comparison

Main results. In Table 1, TaxoIndex performs better than all baselines on both backbone models across various metrics. Notably, TaxoIndex consistently outperforms FFT despite using significantly fewer trainable parameters, and aFT despite using the same add-on module. This shows the efficacy of the proposed approach using the semantic index. Conversely, GRF often degrades performance. The LLM-generate contexts are not tailored to target documents, potentially causing discrepancies in expressions and focused aspects.[7] JTR also fails to outperform FFT. It relies on document clustering, which may be less effective in specialized domains. Among the baselines, ToTER shows competitive performance by leveraging topic information. However, it cannot consider fine-grained concepts not covered by topics, and adds topic information on-the-fly only at inference, failing to fully enhance the backbone retriever. Lastly, while TaxoIndex ++ brings improvements, they are not significantly high, possibly because the phrase information is already reflected by TaxoIndex.

For the subsequent analyses, we use SPECTER-v2 for CSFCube and Contriever-MS for DORIS-MAE which show the highest NDCGs in Table 1.

Impacts of training data. Table 2 reports the improvements by FFT and TaxoIndex with limited

Training data		CSFCube		DORIS-MAE	
		N@5	R@50	N@5	R@50
100%	FFT	+5.92%	+9.84%	+6.09%	+0.59%
	TaxoIndex	**+30.36%**†	**+20.73%**†	**+7.91%**†	**+9.74%**†
50%	FFT	+0.80%	+7.91%	+3.46%	- 0.12%
	TaxoIndex	**+20.65%**†	**+19.26%**†	**+7.58%**†	**+8.40%**†
10%	FFT	+0.51%	+0.40%	+2.29%	- 0.41%
	TaxoIndex	**+19.63%**†	**+15.64%**†	**+6.76%**†	**+7.69%**†

Table 2: Results with varying amounts of training data. We report improvements over no Fine-Tuning. † denotes $p < 0.05$ from paired t-test with FFT.

[6] The module accounts for 6.7% of the model parameters.

[7] For example, for the query in Figure 1, 'simulation-based learning' is included in the generated topics, which may be related but not covered by papers in the corpus.

		(a) High lexical mismatch		(b) High concept diversity	
		N@5	R@50	N@5	R@50
CSFCube	FFT	+5.81%	- 1.45%	+5.01%	+0.78%
	ToTER	+16.19%	+11.86%	+5.10%	+10.90%
	TaxoIndex	**+56.35%***	**+16.94%***	**+50.22%***	**+11.25%**
DORIS-M.	FFT	+1.04%	+3.47%	+0.10%	- 2.21%
	ToTER	+1.07%	+14.24%	+6.90%	+4.78%
	TaxoIndex	**+8.58%***	**+24.65%***	**+10.41%***	**+11.16%***

Table 3: Further analysis for *difficult queries*. We report improvement over no Fine-Tuning. * denotes $p < 0.05$ from paired t-test with ToTER.

training data. FFT shows restricted improvements with limited data and even degrades performance. In contrast, TaxoIndex consistently achieves significant improvements, even with highly limited training data. This is practically advantageous for real-world applications where collecting ample data is challenging. TaxoIndex trains the model to explicitly identify the most important concepts based on the index knowledge, effectively reducing reliance on the training data.

Difficult query analysis. In Table 3, we further analyze results for *difficult queries*, which account for 20% of total test queries. They are identified by two factors complicating query comprehension: (a) high lexical mismatch with documents, (b) high concept diversity within the query.[8] FFT shows limited results and even degrades performance, despite the overall improvement in Table 1. Conversely, TaxoIndex consistently improves the retrieval quality on both types of queries, effectively handling lexical mismatch and various academic concepts. These results in §5.2 collectively show the effectiveness of TaxoIndex in academic paper search.

5.3 Study of TaxoIndex

Ablation study. Table 4 presents various ablation results. First, the best performance is achieved by indexing both topics and phrases. Notably, removing phrase information drastically degrades performance, as phrases enable fine-grained distinctions of each document. Conversely, the absence of topic-level can be partially compensated by phrases, leading to smaller performance drops. Second, both architecture choices improve the indexing network, verifying the efficacy of leveraging the complementarity of topics and phrases and the topic hierarchy. Lastly, both adaptive weight and topic-aware mining prove effective. The mining technique shows

[8]We select queries with (a) the lowest BM25 scores for the top-5% retrieved documents, (b) the highest average pair-wise distance among core topic embeddings of relevant documents.

	N@5	N@10	R@50
TaxoIndex	**0.458**	**0.417**	**0.633**
Indexed information			
w/o Topic-level	0.415	0.382	0.619
w/o Phrase-level	0.385	0.369	0.578
Indexing network architecture			
w/o Multi-gate mixture of experts	0.415	0.380	0.626
w/o GNN	0.420	0.402	0.620
Training technique			
w/o Input-adaptive weight (Eq.3)	0.430	0.404	0.631
w/o Topic-aware negative mining	0.432	0.403	0.630
aFT (w/o index-grounded fine-tuning)	0.378	0.344	0.578

Table 4: Ablation study on CSFCube.

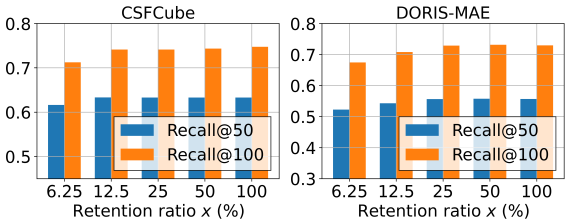


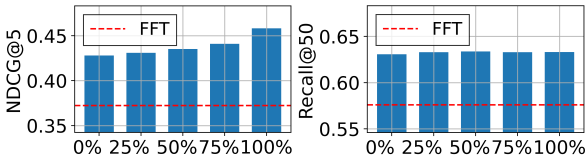
Figure 4: Results with varying retention ratio x.

higher impacts on top-ranked documents and often leads to faster convergence in our experiments.

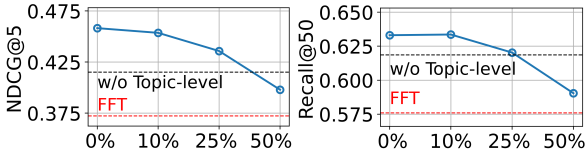
Document Filtering based on Core Topics. Figure 4 shows the retrieval performance with varying retention ratios. We observe that topic-based filtering achieves comparable results to a whole corpus search by examining about 25% of the documents. This result indicates that core topics indeed effectively capture the central theme of each document.

We expect that core topics can be leveraged to improve recent clustering-based ANN indexes (Zhan et al., 2022; Li et al., 2023), which conduct clustering on document embeddings and use cluster memberships to represent documents. As topics are already discrete categories, this approach can reduce the need for clustering operations and provide guidance during the clustering process. Additionally, it offers interpretability by explicitly using topic names. As this is not the focus of our work, we leave further investigation for future research.

Impact of LLM-based topic filtering. For core topic identification, TaxoIndex utilizes LLM-based filtering (§4.1.1). In Figure 5(a), we explore its impacts by replacing it with score-based filtering, which retains documents with similarity above the median similarity of all documents assigned to each topic. Both score- and LLM-based filtering consistently achieve significant improvements over FFT. LLM-based filtering discerns detailed topics bet-



(a) Results with varying LLM-based topic filtering ratios.



(b) Results with varying taxonomy impairment ratios.

Figure 5: Taxonomy-related analysis on CSFCube.

ter, aiding in finding more precise order among the top-ranked documents, and further enhancing interpretability. This result also supports the effectiveness of our candidate topic generation strategy.

Impact of taxonomy quality. TaxoIndex utilizes an academic taxonomy to guide index construction. In Fig.5(b), we explore the impact of taxonomy quality by impairing completeness through random pruning, controlling the ratio of removed nodes to total nodes. TaxoIndex shows considerable robustness, outperforming FFT even with 50% pruning. During training, missing topics can be partially inferred from existing topics and phrases, and detailed phrase information can compensate for incompleteness not covered by the topics. This analysis shows that TaxoIndex is not heavily dependent on taxonomy quality. We expect TaxoIndex to be effective with taxonomies available on the web, and further improved with existing taxonomy completion techniques (Lee et al., 2022; Xu et al., 2023; Zhang et al., 2024).

6 Conclusion

We propose TaxoIndex to match academic concepts in paper search effectively. TaxoIndex extracts key concepts from papers and constructs a semantic index guided by an academic taxonomy. It then trains an add-on module to identify and incorporate these concepts, enhancing dense retrievers. Extensive experiments show that TaxoIndex yields significant improvements, even with limited training data.

We expect that TaxoIndex will effectively improve retrieval quality in various domains where underlying search intents are not sufficiently revealed by surface text. Specifically, e-commerce is an interesting and promising domain. In this domain, users often express their information needs in various forms rather than searching by the exact product name. They might include desired attributes, characteristics, or even specific use cases. TaxoIndex can be applied to better capture users' search intents in such scenarios. We leave further investigation as future work.

7 Limitations

Despite the satisfactory performance of TaxoIndex, our study has three limitations.

First, we utilize an academic taxonomy obtained from the web to guide core topic identification (§4.1.1). We acknowledge that the taxonomy may not reflect up-to-date information. However, we are optimistic that this issue can be addressed by leveraging automatic taxonomy construction and completion techniques, a well-established research fields with many readily available tools (Lee et al., 2022; Xu et al., 2023; Shi et al., 2024). Also, our analysis in §5.3 shows that TaxoIndex has considerable robustness to taxonomy coverage by utilizing phrase information directly extracted from papers.

Second, for topics and phrase mining process (§4.1), we employ relatively simple techniques (e.g., distinctiveness and integrity computations) that have proven effective in recent text mining work. While these choices show high effectiveness in our experiments, we acknowledge that more sophisticated techniques could be employed. Our primary contributions lie in representing each paper's concepts at two levels and incorporating them into relevance predictions, rather than in the specific details for obtaining topics and phrases.

Lastly, this work focuses on the typical dense retrieval models that represent each text as a single vector embedding. Applying TaxoIndex to multi-vector representation models (Santhanam et al., 2022) may require additional modifications, which have not been explored in this study.

8 Ethical Statement

We utilize widely recognized and publicly available datasets for research purposes. Our methodologies and findings do not cause harm to any individuals or groups. We do not foresee any significant ethical issues arising from our work.

Acknowledgements

This work was supported IITP grant funded by MSIT (No.2018-0-00584, No.2019-0-01906), NRF grant funded by the MSIT (No.RS-2023-00217286, No.2020R1A2B5B03097210). It was also in part

by US DARPA INCAS Program No. HR0011-21-C0165 and BRIES Program No. HR0011-24-3-0325, National Science Foundation IIS-19-56151, the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329.

References

Dmitry Baranchuk, Artem Babenko, and Yury Malkov. 2018. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In *ECCV*, pages 202–216.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *EMNLP*, pages 3615–3620.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient inverted indexes for approximate retrieval over learned sparse representations. In *SIGIR*.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *SIGIR*, pages 2353–2359.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *ACL*, pages 2843–2853.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

SeongKu Kang, Shivam Agarwal, Bowen Jin, Dongha Lee, Hwanjo Yu, and Jiawei Han. 2024. Improving retrieval in theme-specific applications using a corpus topical taxonomy. In *WWW*, page 1497–1508.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. Taxocom: Topic taxonomy completion with hierarchical discovery of novel topic clusters. In *WWW*, pages 2819–2829.

Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. 2023. Constructing tree-based index for efficient and effective dense retrieval. In *SIGIR*, pages 131–140.

Xiao Liu, Da Yin, Jingnan Zheng, Xingjian Zhang, Peng Zhang, Hongxia Yang, Yuxiao Dong, and Jie Tang. 2022. Oag-bert: Towards a unified backbone language model for academic knowledge services. In *KDD*, page 3418–3428.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, pages 1930–1939.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *SIGIR*, page 2026–2031.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *ACL*, pages 4089–4100.

Edleno Silva de Moura and Marco Antonio Cristo. 2009. *Indexing the Web*, pages 1463–1467. Springer US, Boston, MA.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *NAACL*, pages 4453–4470.

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. Csfcube-a test collection of computer science research articles for faceted query by example. *NeurIPS 2021 Track on Datasets and Benchmarks*.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In *EMNLP*.

Ninh Pham and Tao Liu. 2022. Falconn++: A locality-sensitive filtering approach for approximate nearest neighbor search. In *NeurIPS*, pages 31186–31198.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL-HLT*, pages 5835–5847.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *NAACL*, pages 3715–3734.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A web-scale system for scientific knowledge exploration. In *Proceedings of ACL 2018, System Demonstrations*, pages 87–92.

Jingchuan Shi, Hang Dong, Jiaoyan Chen, Zhe Wu, and Ian Horrocks. 2024. Taxonomy completion via implicit concept insertion. In *WWW*, pages 2159–2169.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. Scirepeval: A multi-format benchmark for scientific document representations. In *EMNLP*, pages 5548–5566.

Fangbo Tao, Honglei Zhuang, Chi Wang Yu, Qi Wang, Taylor Cassidy, Lance M Kaplan, Clare R Voss, and Jiawei Han. 2016. Multi-dimensional, phrase-based summarization in text cubes. *IEEE Data Eng. Bull.*, 39(3):74–84.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks Track*.

Jianyou Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. 2023. Scientific document retrieval using multi-level aspect-based queries. In *NeurIPS Datasets and Benchmarks Track*.

Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. Pseudo-relevance feedback for multiple representation dense retrieval. In *SIGIR*, pages 297–306.

Hongyuan Xu, Ciyi Liu, Yuhang Niu, Yunong Chen, Xiangrui Cai, Yanlong Wen, and Xiaojie Yuan. 2023. Tacoprompt: A collaborative multi-task prompt learning method for self-supervised taxonomy completion. In *EMNLP*, pages 15804–15817.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *CIKM*, pages 3592–3596.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR*, pages 1503–1512.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Learning discrete representations via constrained clustering for effective and efficient dense retrieval. In *WSDM*, pages 1328–1336.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker model for dense retrieval. In *ICLR*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Zhicheng Dou, and Jing Yao. 2023a. Hybrid inverted index is a robust accelerator for dense retrieval. In *EMNLP*.

Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023b. Pre-training multi-task contrastive learning models for scientific literature understanding. In *Findings of EMNLP*, pages 12259–12275.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2024. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. *arXiv preprint arXiv:2403.00165*.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. Bert-qe: Contextualized query expansion for document re-ranking. In *Findings of EMNLP*, pages 4718–4728.

A Prompt for Core Topic Selection

We instruct LLMs using the prompt provided below. Both datasets used in this work contain paper abstracts. In our experiments, the average number of candidate topics is 28.5, and the average number of selected topics is 9.4.

It is important to note that representing the vast number of nodes in the taxonomy within a single prompt is infeasible. Our two-step strategy, which first identifies candidate topics and then pinpoints core topics, facilitates effective core topic selection.

Instruction: You will receive a paper abstract along with a set of candidate topics for the paper. Your task is to select the topics that best align with the core theme of the paper. Exclude topics that are too broad or less relevant. You may list up to 10 topics, using only the topic names in the candidate set. Do not include any explanation.
Paper: [DOCUMENT],
Candidate topic set: [CANDIDATE TOPICS]

B Details of Experiment Setup

B.1 Dataset

We have surveyed the literature to find retrieval datasets in the academic domain where relevance is labeled by experts (or annotators with high capabilities). We select two recently published datasets: CSFCube (Mysore et al., 2021) and DORIS-MAE (Wang et al., 2023). They provide test query collections along with relevance labels, annotated by human experts and LLMs, respectively. They also represent two real-world search scenarios: query-by-example and human-written queries.

For both datasets, we conduct retrieval from the entire corpus including all candidate documents. CSFCube dataset consists of 50 test queries, with about 120 candidates per query drawn from approximately 800,000 papers in the S2ORC corpus. Annotation scores greater than '2' (nearly identical or similar) are treated as relevant. We use the title as the query and both the title and abstract for the documents. DORIS-MAE dataset consists of 100 test queries, with about 100 candidates per query drawn similarly to CSFCube dataset. For each query, average annotation scores greater than '1' (the document answers some or all key components) are treated as relevant.

Lastly, we provide results of Contriever-MS on SCIDOCS (Cohan et al., 2020; Thakur et al., 2021) in Appendix C.1. SCIDOCS dataset contains $1,000$ test queries with relevance labels for $25,657$ papers. Please note that we exclude this dataset from main experiments, as it uses citation relations for relevance labels, which are utilized for the training of SPECTER-v2.

B.2 Academic Taxonomy

We use the field of study from Microsoft Academic (Shen et al., 2018), which covers 19 disciplines (e.g., computer science, biology). It contains $431,416$ nodes and $498,734$ edges with a maximum depth of 4. Note that we prune the taxonomy by only retaining topics included in the target corpus, during the indexing process (§4.1.1). The number of nodes after the pruning is provided in §5.1.

B.3 Metrics

Following the previous work (Thakur et al., 2021; Mackie et al., 2023; Kang et al., 2024), we employ Recall@K (R@K) for a large retrieval size (K), and NDCG@K (N@K) and MAP@K for a smaller K. Recall@K measures the proportion of relevant documents retrieved in the top K results, without consideration of the rank of the documents. Conversely, NDCG@K and MAP@K directly consider the absolute rank of each relevant document, where a higher value indicates that relevant documents are consistently found at higher ranks.

B.4 Experiment Details

Backbone models. We use publicly available checkpoints: SPECTER-v2[9] and Contriever-MS[10]. SPECTER-v2 is trained via multi-task learning using paper metadata from SCIBERT (Beltagy et al., 2019), and Contriever-MS is fine-tuned via massive training queries (MS MARCO) from BERT base uncased (Devlin et al., 2018). Both backbone models have about 110 million parameters.

Computational resources and API cost. We conduct all experiments using 4 NVIDIA RTX A5000 GPUs, 512 GB of memory, and a single Intel Xeon Gold 6226R processor. For ChatGPT API usage, we spent $11.50 on core topic selection in TaxoIndex and $39.30 on query generation.

Implementation details. For BM25, we use Elasticsearch.[11] As training queries are not provided in all datasets used in this work, we leverage synthetic queries generated by using PROMPTGATOR (Dai et al., 2023), the state-of-the-art query generation method.[12] All compared methods are trained using the same queries. We use 10% of training data as a validation set. We share the generated queries for reproducibility. We report the average performance over five independent runs.

- **Fine-tuning details.** FFT and aFT use top-50 hard negatives mined from BM25 for each query, as done in Formal et al. (2022). TaxoIndex uses top-50 hard negatives mined using core topics and BM25 scores (§4.2.3). We use the inner

[9] `allenai/specter2_base`
[10] `facebook/contriever-msmarco`
[11] https://github.com/elastic/elasticsearch
[12] We use `gpt-3.5-turbo-0125` for query generator.

product as a similarity function for relevance prediction. The learning rate is set to $1e^{-6}$ for FFT and $1e^{-4}$ for aFT and TaxoIndex, after tuning among $\{1e^{-7}, 1e^{-6}, ..., 1e^{-3}\}$. We set the batch size as 128 and the weight decay as $1e^{-4}$.

- **GRF, ToTER, and JTR.** We utilize the official implementation for GRF[13], ToTER[14], and JTR[15]. For GRF, we generate both topics and keywords using the same LLMs with TaxoIndex for a fair comparison. ToTER and TaxoIndex utilize the same given taxonomy. For baseline-specific hyperparameters, we closely follow the recommended values in the original papers and implementations.

- **TaxoIndex.** TaxoIndex only updates an add-on module that has 7.38 million parameters, which account for about 6.72% of backbone models. For core topic selection using LLM, we set the temperature as 0.2. The phrase set $\mathcal{P}$ is obtained using AutoPhrase.[16] In §4.1.2, the number of phrases per document is set as $k = \min(15, P_d \times 0.2)$, where P_d denotes the total number of phrases in the document d. This simple choice allows for a natural consideration of the document length. The average number of indicative phrases is 13.2 for CSFCube and 13.7 for DORIS-MAE. For TaxoIndex ++, we set $k = 15$. We set the size of topically similar document set $|\mathcal{D}_d| = 100$.

  For the indexing network, we set the number of experts as $M = 3$, each being a two-layer MLP. g^t, g^p are linear layers with Softmax outputs. For the topic encoder, we use a two-layer graph convolution network (Kipf and Welling, 2016). For the fusion network, we use a two-layer MLP for f^I, and a linear layer for $g^I : \mathbb{R}^l \rightarrow \mathbb{R}$.

  For index learning of training queries, we use the averaged labels of the relevant documents when a query has multiple relevant documents. We first warm up the indexing network using $\mathcal{L}_{IL}$ until the training loss converges. We set $\lambda_{IL} = 0.1$.

[13] https://drive.google.com/drive/folders/1LWGTvXGatrAbwbDahYkraK-nim2O9eyN

[14] https://github.com/SeongKu-Kang/ToTER_WWW24

[15] https://github.com/cshaitao/jtr

[16] https://github.com/shangjingbo1226/AutoPhrase

C Supplementary Results

C.1 SCIDOCS Results

We provide results of Contriever-MS on SCIDOCS. Please note that we exclude this dataset from the main experiments, as it uses citation relations for relevance labels, which are utilized for the training of SPECTER-v2. We conduct automatic evaluation using LLMs as well as conventional evaluation using relevance labels.

Conventional evaluation. Table 5 presents the retrieval results. Overall, TaxoIndex shows higher retrieval performance compared to FFT and ToTER, despite using significantly fewer trainable parameters.

	N@5	N@10	R@20	R@50	R@100
FFT	- 7.71%	- 5.18%	- 2.15%	+1.41%	+4.47%
ToTER	+4.22%	+6.05%	+9.37%	+10.47%	+14.00%
TaxoIndex	**+13.29%**	**+14.74%**	**+15.15%**	**+16.70%**	**+18.56%**

Table 5: Results on SCIDOCS dataset. We report relative performance change with respect to the backbone retriever (i.e., no Fine-Tuning).

Automatic evaluation. For a more thorough evaluation with explicit consideration of detailed contents, we employ automatic evaluation using LLMs. We adopt a recent pair-wise ranking technique (Qin et al., 2023) that instructs LLMs to compare the relevance of two passages for a given query. The prompt is provided below.

> Given a query [QUERY], which of the following two documents is more relevant to the query?
> Document A: [DOCUMENT A]
> Document B: [DOCUMENT B]
> Output Document A or Document B.

We compare top-1 retrieval results from TaxoIndex and each baseline, applying this evaluation only if the results are not identical. Figure 6 presents the results. Notably, the improvements revealed through automatic evaluation are much larger compared to those from conventional evaluation. These

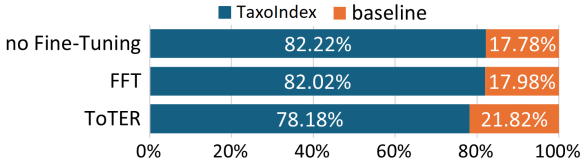


Figure 6: Win ratios of TaxoIndex and each baseline method by automatic evaluation using LLMs. We use `gpt-3.5-turbo-0125`.

results collectively support the effectiveness of TaxoIndex in the academic paper search.

C.2 Indexing Network Performance

Table 6 presents the classification performance of the indexing network. We report the Precision@10 results on the training set, after the warmup of the indexing network. We observe high precision for both topics and phrases, indicating they are well captured by the proposed network.

	CSFCube	DORIS-MAE
Topic-level	0.863	0.865
Phrase-level	0.998	0.980

Table 6: Topic and phrase classification performance of the indexing network.

C.3 Additional Case Study

Setup. As discussed in §4.3, topics and phrases with the highest predicted probabilities reveal academic concepts captured and reflected for retrieval. We interpret search results by comparing predictions for queries $(\hat{\mathbf{y}}_q^t, \hat{\mathbf{y}}_q^p)$ and documents $(\hat{\mathbf{y}}_d^t, \hat{\mathbf{y}}_d^p)$. In our case studies in Figure 1, Table 7, and Table 8, we use topics and phrases having the highest logit values. Note that we use $\hat{\mathbf{y}}$ instead of $\mathbf{y}$ for documents, as unlabeled but relevant classes are naturally revealed during training.

Case study: short query with limited context. In Table 7, we present inferred information for two example queries. For the query 'semantic parsing learning with limited labels', TaxoIndex infers concepts such as 'syntactic predicate' and 'semi-supervised learning'. Similarly, for the query 'domain adaptation approach for machine translation', TaxoIndex identifies related concepts like 'parallel corpus' and 'NMT (neural machine translation)'. This inferred information complements limited query context, facilitating concept matching for paper search. We highlight that our index is constructed by organizing knowledge in the target corpus, and thus these terminologies are actually used in the papers that users search.

Case study: long and complex query. In Table 8, we explore how TaxoIndex handles long and complex queries by analyzing one that includes various concepts. For this query, we present retrieval results: (a) an easy case that is well handled by all baselines (document A), and (b) two difficult cases that are not effectively handled by baselines (documents B and C).

The query encompasses various academic concepts: generative approaches for creating game levels, optimization via reinforcement learning or other differentiable methods, and measuring agent performance. Document A is ranked at the top-1 by all compared methods due to its high lexical overlap, directly including terms used in the query (e.g., GAN, generated levels). In contrast, documents B and C are not retrieved near the top. Unlike document A, they express the concepts using different terms (e.g., neuroevolutionary system), making it difficult to find relevance using surface texts. Additionally, document C specifically focuses on surrogate models for a shooter game, which obscures the query concepts like level generation.

TaxoIndex infers the most relevant topics and phrases from the query (highlighted in yellow) and incorporates them into relevance prediction. This helps to match the underlying academic concepts, improving retrieval results. However, it still shows limited effectiveness for document C, as the overlap of indexed information is relatively small. We also note that fine-grained aspects of 'surrogate model' and 'character class' are not fully included in the indexed information, potentially because there are fewer documents covering such concepts in the corpus. We expect that incorporating other knowledge sources (e.g., knowledge bases) can mitigate these problems. We leave further exploration for future work.

Query
semantic parsing learning with limited labels
Inferred core topics and indicative phrases
parsing, labeled data, statistical parsing, parser combinator, top down parsing, syntactic predicate, text annotation, semi-supervised learning, morphological parsing, natural language processing, artificial intelligence
semantic parsing, semantic parsers, taggers, syntactic parsing, parse tree, semantic role labeling, treebanks, predicates, semantic representations, formal semantics, predicate argument, linguistic knowledge, training labels, ...
Query
domain adaptation approach for machine translation
Inferred core topics and indicative phrases
domain adaptation, machine translation, semantic textual similarity, semantic translation, translation probabilities, weakly supervised learning, neural machine translation, natural language processing, machine learning, artificial intelligence
neural machine translation, statistical machine translation, nmt, smt, cross sentence, adaptation techniques, domain adaptation, parallel sentences, parallel corpus, out of domain, model adaptation, translators, machine translation, ...

Table 7: Case study for queries having limited contexts (Corpus: CSFCube)

Query

I am seeking a generative modeling approach capable of creating new levels and potentially game settings/environments for a video game with multiple existing levels of difficulty. Specifically, I am interested in exploring how Generative Adversarial Networks (GANs) and other generative methods could generate entirely new levels by emulating the style of previous ones. It is crucial that the newly generated levels are not merely derivative and that my generative model can optimize specific properties, such as the intensity or graphic nature of the game. Given that these properties are non-differentiable, I need a method to either render them differentiable or employ a reinforcement learning-centric approach to optimize these rewards. After generating a variety of levels, I require a method to select some of the best ones. One potential solution could be to evaluate the generated levels using an automatic metric, such as the performance of an AI agent playing the level. Alternatively, I am considering designing a derivative-free stochastic optimization algorithm to guide the search across the space of all synthetically generated levels, steering towards those that meet specific objectives.

Document A: an *easy* case (Top-1 by all compared methods)

Illuminating mario scenes in the latent space of a generative adversarial network. Generative adversarial networks (GANs) are quickly becoming a ubiquitous approach to procedurally generating video game levels. While GAN generated levels are stylistically similar to human-authored examples, human designers often want to explore the generative design space of GANs to extract interesting levels. However, human designers find latent vectors opaque and would rather explore along dimensions the designer specifies, such as number of enemies or obstacles. We propose using state-of-the-art quality diversity algorithms designed to optimize continuous spaces, i.e. MAP-Elites with a directional variation operator and Covariance Matrix Adaptation MAP-Elites, to efficiently explore the latent space of a GAN to extract levels that vary across a set of specified gameplay measures. In the benchmark domain of Super Mario Bros, we demonstrate how designers may specify gameplay measures to our system and extract high-quality (playable) levels with a diverse range of level mechanics, while still maintaining stylistic similarity to human authored examples. An online user study shows how the different mechanics of the automatically generated levels affect subjective ratings of their perceived difficulty and appearance.

Document B: a successful *difficult* case (FFT: top-62, ToTER: top-39, TaxoIndex: top-9)

Co-generation of game levels and game-playing agents. Open-endedness, primarily studied in the context of artificial life, is the ability of systems to generate potentially unbounded ontologies of increasing novelty and complexity. Engineering generative systems displaying at least some degree of this ability is a goal with clear applications to procedural content generation in games. The Paired Open-Ended Trailblazer (POET) algorithm, heretofore explored only in a biped walking domain, is a coevolutionary system that simultaneously generates environments and agents that can solve them. This paper introduces a POET-Inspired Neuroevolutionary System for KreativitY (PINSKY) in games, which co-generates levels for multiple video games and agents that play them. This system leverages the General Video Game Artificial Intelligence (GVGAI) framework to enable co-generation of levels and agents for the 2D Atari-style games Zelda and Solar Fox. Results demonstrate the ability of PINSKY to generate curricula of game levels, opening up a promising new avenue for research at the intersection of procedural content generation and artificial life. At the same time, results in these challenging game domains highlight the limitations of the current algorithm and opportunities for improvement.

Indexed information (Inferred from the query)

video game development, heuristics, game design, intelligent agent, genetic algorithm, reward technique, optimization problem, evolutionary algorithm, reinforcement learning, heuristic evaluation, generative model, machine learning, artificial intelligence

game playing, game levels, video games, game design, agents, players, general video game ai, exploration and exploitation, GVGAI, simultaneously learn, optimization, autonomous agents, content generation, knowledge acquisition, rewards, ...

Document C: an unsuccessful *difficult* case (FFT: top-258, ToTER: top-155, TaxoIndex: top-108)

Pairing character classes in a deathmatch shooter game via a deep-learning surrogate model. This paper introduces a surrogate model of gameplay that learns the mapping between different game facets, and applies it to a generative system which designs new content in one of these facets. Focusing on the shooter game genre, the paper explores how deep learning can help build a model which combines the game level structure and the game's character class parameters as input and the gameplay outcomes as output. The model is trained on a large corpus of game data from simulations with artificial agents in random sets of levels and class parameters. The model is then used to generate classes for specific levels and for a desired game outcome, such as balanced matches of short duration. Findings in this paper show that the system can be expressive and can generate classes for both computer generated and human authored levels.

Indexed information (Inferred from the query)

simulation, surrogate model, game design, game mechanics, modeling and simulation, parameter, video game development, intelligent agent, reinforcement learning, generative model, machine learning, deep learning, artificial intelligence

surrogate model, artificial agents, game data corpus, simulation, environments, game levels, game playing, content generation, agents, players, characters, character levels, general video game ai, parameters, game design, video games, generators, ...

Table 8: Case study for a long and complex query. All documents are labeled as relevant (Corpus: DORIS-MAE).