CASPER: Concept-integrated Sparse Representation for Scientific Retrieval

Anonymous ACL submission

Abstract

Identifying relevant research concepts is crucial for effective scientific search. However, primary sparse retrieval methods often lack concept-aware representations. To address this, we propose CASPER, a sparse retrieval model for scientific search that utilizes both tokens and keyphrases as representation units (i.e., dimensions in the sparse embedding space). This enables CASPER to represent queries and documents via research concepts and match them at both granular and conceptual levels. Furthermore, we construct training data by leveraging abundant scholarly references (including titles, citation contexts, author-assigned keyphrases, and co-citations), which capture how research concepts are expressed in diverse settings. Empirically, CASPER outperforms strong dense and sparse retrieval baselines across eight scientific retrieval benchmarks. We also explore the effectiveness-efficiency trade-off via representation pruning and demonstrate CASPER's interpretability by showing that it can serve as an effective and efficient keyphrase generation model.

1 Introduction

Science progress is accelerating at an unprecedented rate, with millions of research articles published yearly (Fortunato et al., 2018). This surge in scientific output makes it increasingly difficult for researchers to catch up with the latest developments (Huettemann et al., 2025). As a result, there is a pressing need for robust information retrieval systems that can efficiently navigate and filter the vast landscape of scientific literature.

To address this challenge, the community has largely turned to Dense Retrieval (DR) models (Cohan et al., 2020; Singh et al., 2023; Mysore et al., 2022; Kang et al., 2024, 2025), which encode documents into continuous vector spaces to capture semantic nuances. However, Learned Sparse Retrieval (LSR) has recently emerged as an attractive alternative. Unlike DR models, LSR models encode text into sparse lexical vectors, where each term (word or subword) serves as a representation unit, i.e., a dimension in the embedding space. The advantages of LSR over dense retrieval include 1) efficiency due to its compatibility with inverted indexes and 2) interpretability since each dimension in the representation space corresponds to a term. Realizing the full potential of LSR in this context of scientific document search, however, requires overcoming three challenges.

Query	deep transfer learning in neural networks
SPLADE	(deep, 2.55), (transfer, 2.27), (neural, 1.73), (learning, 1.56), (brain, 1.37), (networks, 1.33), (learn, 1.26), (network, 1.15), (transferred, 1.14), (college, 1.03), (,, 0.69), (goal, 0.61), ...
CASPER	(deep learning, 2.26), (deep, 1.43), (knowledge transfer, 1.36), (transfer learning, 1.25), (soap, 1.24), (neural network, 1.15), ..., (domain adaptation, 0.55), ..., (artificial intelligence, 0.41), ...

Table 1: SPLADE and CASPER representations for a hypothetical query.

The **first challenge** is *the lack of conceptual representation units*. Primary sparse retrieval approaches (Bai et al., 2020; Zhao et al., 2020; Formal et al., 2021b,a) utilize BERT tokens as representation units. However, it has been pointed out that searching for scientific article heavily involves identifying research concepts to include for querying (Bramer et al., 2018). Although tokens allow for matching of granular details, the lack of conceptual representation unit hinder capturing research concepts, potentially causing the model to miss relevant documents. For instance, consider the example query in Table 1. Other than documents that discuss "transfer learning", it is expected that those discussing relevant concepts, such as "knowledge transfer" and "domain adaptation", are also retrieved. While SPLADE (Formal et al., 2021b,a), a prominent sparse model, expands the query representation beyond the present tokens to include related ones. Consequently, SPLADE's representation fails to capture the conceptual relationship and miss relevant articles.

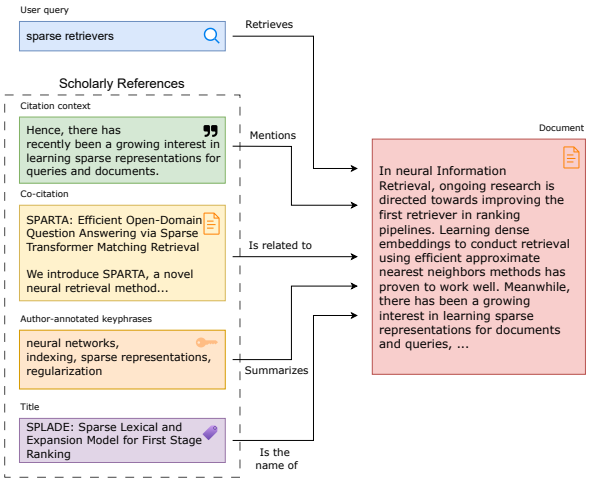

Figure 1: Example of scholarly references

Our key idea for addressing this challenge is to *learn concept-integrated representation*. In particular, we propose to treat research concepts as representation units in addition to tokens, allowing matching both token and concept-level details.

The **second challenge** is that it is unclear regarding the criterion to *determine conceptual representation units*. We propose *selecting common and comprehensive keyphrase vocabulary as concept units*. Keyphrases are typically used to describe research concepts and therefore are a natural choice for concept units. The selected keyphrases should be common, widely present in queries and documents, to ensure sufficient instances for each dimension for the model to learn. Furthermore, the keyphrase vocabulary should be comprehensive, covering the diverse research concepts circulating in the literature, enabling each document to be represented with these keyphrases. For instance, a vocabulary limited to Computer Science terminology cannot effectively represent Biology articles.

The **third challenge** is *limited supervision signals*. Existing data sources do not support learning concept-aware representation. Previous work utilizes pairs of articles, formed by either citation links (Cohan et al., 2020; Singh et al., 2023) or co-citations (Mysore et al., 2022). However, besides the potential inaccuracies of these signals (as in the case of citation links (Mysore et al., 2022)), relying solely on them is insufficient. More specifically, they mainly indicate related concepts mentioned in *other works*, but do not directly reveal the concepts expressed in the given text itself.

User queries and interaction are another potential source of supervision, with an example being SciRepEval Search (Singh et al., 2023). However, such signals are typically proprietary and difficult to obtain. Moreover, they are also insufficient. Our analysis reveals that a significant portion of queries in SciRepEval Search closely match paper titles. In particular, 44% of query–document pairs involve queries that appear as substrings in the document title or vice versa, and 53% contain all query terms within the document title. This pattern indicates predominantly known-item rather than exploratory, concept-driven searches. It is noteworthy that this reflects the limitations of current scholarly search engines, which primarily support known-item retrieval (Gusenbauer and Haddaway, 2021), rather than scientific document search being a simple task.

Our key idea for solving this challenge is *leveraging diverse scholarly references*. We refer to scholarly references as signals within the literature to connect and describe academic articles *at the concept level*. We specifically utilize four sources namely *titles*, *citation contexts*, *author-assigned keyphrases*, and *co-citation networks* as (pseudo) queries to retrieve the underlying referenced documents. These signals are embedded within the scientific literature and are therefore "free" to obtain. In addition, they capture research concepts of papers expressed in different settings and therefore are rich sources of supervision. In Figure 1, we provide an example of these sources of supervision.

We summarize our contributions as follows. **Firstly**, we propose **CASPER**, a sparse retrieval model that utilizes both tokens and keyphrases as representation units. **Secondly**, we introduce a framework to construct large-scale scientific Information Retrieval (IR) training data by mining scholarly references, which we name **FRIEREN**. **Finally**, we extensively evaluate CASPER on eight scientific retrieval benchmarks, where it outperforms strong dense and sparse baselines. We also explore the effectiveness-efficiency trade-off via representation pruning and demonstrate CASPER's interpretability by showing its utility as a robust keyphrase generation model.

2 Related work

Learned Sparse Retrieval. Sparse retrieval models encode inputs as term-based vectors. Traditional methods, such as TF-IDF (Salton and Buckley, 1988) and BM25 (Robertson et al., 2009), generate bag-of-words representations from document and corpus statistics. However, these approaches suffer from *vocabulary mismatching* problem. To

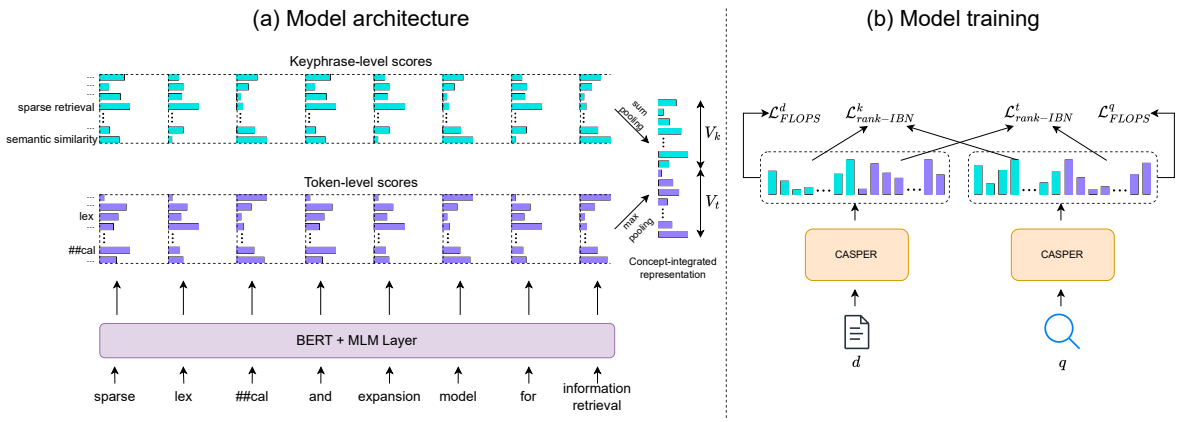

Figure 2: Overview of CASPER, our proposed method.

address this, primary methods (Bai et al., 2020; Dai and Callan, 2019; Zhao et al., 2020; Formal et al., 2021b,a) leverage pretrained language models to learn contextualized sparse representations. By assigning importance scores to tokens from the entire vocabulary, rather than limiting them to those present in the input, these methods effectively mitigate vocabulary mismatch.

Most sparse retrieval models represent text in BERT's vocabulary. Expanded-SPLADE (Dudek et al., 2023) attempts to replace this vocabulary with a customized vocabulary of the 300k most frequent English unigrams. DyVo (Nguyen et al., 2024) enhances BERT's vocabulary by incorporating entities retrieved from a knowledge base. Our proposed method, is more related to DyVo. However, while CASPER learns an end-to-end representation integrating both tokens and keyphrases as representation units, DyVo augments a token-based representation with a frozen entity retrieval system, meaning the association between text and (implicit) entities is fixed and not learned from training data.

Retrieval in the Scientific Domain. Information retrieval in the scientific domain has attracted increasing attention in recent years. Early work (El-Arini and Guestrin, 2011) proposed recommendation methods for scientific articles using features such as citation links and shared authorship. Motivated by the multifaceted nature of scientific texts, aspect-based representations have been explored (Jain et al., 2018; Neves et al., 2019; Chan et al., 2018; Kobayashi et al., 2018; Mysore et al., 2021a). Another line of research emphasizes the importance of research concepts-aware representation (Kang et al., 2024, 2025), to which CASPER belongs. Building upon the foundations laid by prior studies, our work introduces two key distinctions: 1) we propose a sparse retrieval model, whereas existing work focuses on dense retrieval; 2) CASPER is trained end-to-end and learns to discover concepts directly from data, while prior methods rely on external systems (such as LLMs) for their concept representation.

Regarding the supervision signals to train scientific IR models, existing research have introduced citation-based document representations (Cohan et al., 2020; Singh et al., 2023), where models are trained on pairs of papers linked by a citation. However, citation links can be a noisy proxy for semantic relevance, therefore Mysore et al. (2021a) propose co-citation, a more accurate source of supervision. Another source is user queries and interaction data, exemplified by SciRepEval Search (Singh et al., 2023). Our work leverages *scholarly references*, which we define as signals within the literature that connect and describe academic articles at the concept level. These signals are not only easy to obtain, but are also rich as they capture research concepts of articles in different settings.

Keyphrase Generation. Keyphrase generation involves producing phrases that capture the core concepts of a text. Prior work typically formulated as a supervised sequence-to-sequence learning task (Meng et al., 2017). Although keyphrases have proven useful for scientific retrieval (Boudin et al., 2020), focused research on keyphrase generation to enhance retrieval remains limited. Notable exceptions explore the use of keyphrases for query and document expansion (Wu et al., 2022; Do et al., 2025). Our work lies in the intersection of keyphrase generation and information retrieval. Specifically, CASPER is a sparse retrieval model that explicitly represents documents and queries using keyphrases, thereby integrating the capabilities

of a keyphrase generation model within a retrieval framework.

3 Preliminary: SPLADE

Architecture. SPLADE (Formal et al., 2021b,a) is a sparse represetation method that predicts term importances for query/document in the BERT (Devlin et al., 2019) token vocabulary space V_t. In particular, given an input document d, let w_{ij}^d denotes the importance of token $j \in V_t$ given token $i \in d$, predicted by the Masked Language Modeling (MLM) layer. The importance w_j^d of j given d is computed with max pooling

$$w_j^d = \max_{i \in d} \log(1 + \text{ReLU}(w_{ij}^d)) \quad (1)$$

Training. Training SPLADE requires a collection of triplets (q_i, d_i^+, d_i^-), where q_i, d_i^+, d_i^- respectively denotes query, positive document and hard negative document; the set of in-batch negatives is denoted as $\{d_{i,j}^-\}_j$. SPLADE is trained by optimizing a contrastive loss, with both in batch and hard negatives; along with sparsity regularization

$$\mathcal{L} = \mathcal{L}_{rank-IBN} + \lambda_d \mathcal{L}_{FLOPS}^d + \lambda_q \mathcal{L}_{FLOPS}^q \quad (2)$$

$$\mathcal{L}_{rank-IBN} = -\log \frac{e^{s(q_i, d_i^+)}}{e^{s(q_i, d_i^+)} + e^{s(q_i, d_i^-)} + \sum_j e^{s(q_i, d_{i,j}^-)}} \quad (3)$$

where $s(q, d) = \sum_{j \in V_t} w_j^d \; w_j^q$ is the similarity between query q and document d, defined by the dot product of their representation. $\mathcal{L}_{FLOPS}$ is the FLOPS regularizer to enforce sparsity, first introduced in (Paria et al., 2020). For further details on SPLADE, we encourage readers to refer to the original papers (Formal et al., 2021b,a).

4 CASPER

In this section, we detail CASPER, a **C**oncept-integr**A**ted **SP**ars**E** **R**epresentation model for scientific search; an overview is presented in Figure 2.

4.1 Keyphrase Vocabulary

CASPER representation units include tokens and keyphrases. For tokens, we follow previous work and use BERT vocabulary. For keyphrases, we build a common and comprehensive keyphrase vocabulary, which we detail in this section.

Scientific corpus. We start with the S2ORC corpus (Lo et al., 2020), which include 81.1M academic papers, spanning across multiple disciplines. From this corpus[1], we sample 10M papers, using the concatenated title and abstract of each as the source text. We denote the resulting collection as $\mathcal{D} = \{d_1, d_2, ...\}$.

Extracting keyphrases from corpus texts. To ensure building a vocabulary containing common, widely used keyphrases, we extract them from corpus texts using a keyphrase extraction algorithm. In particular, we employ ERU-KG (Do et al., 2025), which is not only effective but also time-efficient (an attribute that is important in our case as there are millions of articles to process). Using the extraction mode of ERU-KG[2], we extract a set of keyphrases $\mathcal{K}_{d_i}$ for each document $d_i \in \mathcal{D}$.

Forming the vocabulary. We aim at a keyphrase vocabulary that is comprehensive, so that documents can be effectively represented with these keyphrases. To achieve this, we model vocabulary construction as an instance of the *maximum coverage problem* (Nemhauser et al., 1978; Hochbaum, 1997). After obtaining $\mathcal{K}_{d_i}$ for all $d_i \in \mathcal{D}$, we form a *document set* $\mathcal{D}_k = \{d_i \in \mathcal{D} \mid k \in \mathcal{K}_{d_i}\}$ for each keyphrase k. From the set of all candidate keyphrases, $\mathcal{K} = \bigcup_{d_i \in \mathcal{D}} \mathcal{K}_{d_i}$, our goal is to select a subset $V_k \subset \mathcal{K}$ of predetermined size $|V_k|$ that solves the following optimization problem

$$\max \quad |\bigcup_{k \in V_k} \mathcal{D}_k| \quad (4)$$

where $|V_k|$ is a hyperparameter and $|V_k| \ll |\mathcal{K}|$.

As this problem is NP-hard (Nemhauser et al., 1978; Hochbaum, 1997), we approximate the solution using a greedy algorithm (Hochbaum, 1997). In particular, V_k is constructed by repeatedly adding the keyphrase that appears in the most previously uncovered documents until it reaches the predetermined size.

Integration of the keyphrase vocabulary. We augment the BERT vocabulary with the keyphrases in V_k, treating each as a new token. Because these newly added keyphrase tokens have randomly initialized embeddings and MLM parameters, we further pretrain the modified BERT model, using the MLM training objective, to help it learn effective representations for the keyphrases. We continue to pretrain the BERT model using the scientific article collection $\mathcal{D}$.

[1] We utilize the pre-processed version provided by sentence-transformers: https://huggingface.co/datasets/sentence-transformers/s2orc

[2] setting $\alpha, \beta = 1$ as in the original paper

4.2 Pooling Strategy

SPLADE leverages max pooling to determine the importance of individual vocabulary tokens, which has been empirically shown to outperform sum pooling in retrieval tasks (Formal et al., 2021a). Nevertheless, this approach may not fully capture the relevance of research concepts. Max pooling excels at identifying the most salient tokens, since a vocabulary token is included in the representation if it receives a strong activation from any input token. However, this does not necessarily align with how research concepts operate within scientific texts, where the relevance of a concept is often indicated by its consistent significance across a document. With this in mind, we propose a hybrid pooling strategy: max pooling for vocabulary tokens to preserve SPLADE's benefits, and sum pooling for concepts to better reflect their importance.

$$w_j^d = \begin{cases} \max_{i \in d} \log(1 + \text{ReLU}(w_{ij}^d)), & \text{if } j \in V_t \\ \sum_{i \in d} \log(1 + \text{ReLU}(w_{ij}^d)), & \text{if } j \in V_k \end{cases} \quad (5)$$

4.3 Training CASPER

Although tokens and keyphrases both act as representation units, they encode different information. Specifically, token-based representations reflect fine-grained distinctions, whereas keyphrase-based representations capture differences in research concepts. This motivates decoupling the training for each representation type, where each type is trained with different negative examples, so that token- and keyphrase-based representations are trained to differentiate on suitable granularity.

Formally, the ranking loss is defined as

$$\mathcal{L}_{rank-IBN} = \lambda^t \mathcal{L}^t_{rank-IBN} + \lambda^k \mathcal{L}^k_{rank-IBN} \quad (6)$$

where $\mathcal{L}^t_{rank-IBN}$, $\mathcal{L}^k_{rank-IBN}$ are ranking losses for training token and keyphrase-based representation, respectively. $\mathcal{L}^t_{rank-IBN}$ is defined the same as in Eq. 3, where the negative documents include both in-batch and hard negatives. In-batch negatives are essentially random negatives and are likely to be different from positive documents in terms of fine-grained details and research concepts. Hard negatives likely share similar research concepts but are different from positive documents in terms of fine-grained details. We define $\mathcal{L}^k_{rank-IBN}$ similar to $\mathcal{L}^t_{rank-IBN}$, but without the hard negative document in the denominator. The reason is that we want keyphrase-based representation to capture differences in research concepts.

4.4 Inference

At inference time, a document d is ranked based on their similarity with query q in terms of fine-grained details and research concepts. Specifically, documents are ranked based on the following function

$$S(q, d) = s^t(q, d) + \beta \, s^k(q, d) \quad (7)$$

where β is a hyperparameter that controls the influence of the concept-level similarity. $s^t(q, d) = \sum_{j \in V_t} w_j^d w_j^q$ and $s^k(q, d) = \sum_{j \in V_k} w_j^d w_j^q$ is the token and concept matching score, respectively.

4.5 CASPER for Keyphrase Generation

Beyond retrieval, CASPER's sparse representations can be adapted for keyphrase generation. By ranking noun phrases extracted from the document against CASPER's sparse weights, we can identify present keyphrases. Simultaneously, highly activated keyphrases $k \in V_k$ in the sparse vector that do not appear in the text serve as absent keyphrases. We provide further details in Appendix A.

5 FRIEREN

In this section, we describe FRIEREN, a framework for mining **FR**ee sc**IE**ntific **RE**trieval supervisio**N**. FRIEREN mines queries from four sources, namely *co-citations*, *citation contexts*, *author-assigned keyphrases* and *titles*. We provide an example of these four sources in Figure 1. Queries mined from these sources are then used to augment user queries, creating a dataset for training. Formally, we aim to build a collection of triplets, denoted as $\mathcal{T} = \{(q_i, d_i^+, d_i^-)\}_{i=1}^{|\mathcal{T}|}$. Here, q_i denotes a query and d_i^+, d_i^- denotes the corresponding positive and hard negative document.

User queries. We utilize SciRepEval Search[3] (Singh et al., 2023). Each query q_i in this set is associated with a list of candidates and their relevance scores. We select candidates whose scores ≥ 1 as d_i^+. Then, a negative document d_i^- is randomly chosen among those whose scores are 0.

Co-citations. Using the full-text S2ORC corpus[4], we identify citation groups within each paper. From each group, we randomly select two articles: one's title serves as the query q_i, and the other's concatenated title and abstract serves as the positive document d_i^+. The hard negative d_i^- is an article

[3] https://huggingface.co/datasets/allenai/scirepeval/viewer/search/train

[4] https://github.com/allenai/s2orc

	SciFact	SCIDOCS	NFCorpus	DORIS-MAE	CSFCube	ACM-CR	LitSearch	RELISH	AVG
NDCG@10									
BM25	67.9	14.9	32.2	20.8	5.8	31.2	40.2	60.9	34.2
SPECTER2	63.9	17.1	24	16.3	5	22.3	33.2	59	30.1
E5-base-v2	**71.4**	**18.9**	**35.5**	16.9	5.3	28.7	40.2	65.3	35.3
DyVo	68.2	15.3	34.5	9.6	**6.5**	26.6	40.5	59.9	32.6
ColBERT	65.7	14.7	30.9	20.1	4.9	28.5	37.6	60.1	32.8
SPLADEv2	63.8	14.9	32	17.2	5.3	27.2	42.8	60.6	33
$\text{ColBERT}_{\text{FRIEREN}}$	70.4	16.8	33	**23.9**	5.7	34	46.2	**66.9**	37.1
$\text{SPLADEv2}_{\text{FRIEREN}}$	**71.4**	16.8	27.1	17.8	4.6	33.8	41.6	65.1	34.8
$\text{SPLADEv2}^{\text{words}}_{\text{FRIEREN}}$	70.5	16.9	28.4	20.8	5.3	32.1	43.8	64.7	35.3
CASPER	70.2	18.8	32.2	21.3	5.5	34.2	46.8	65	36.8
CASPER++	70.6	18.5	34.2	23.7	5.6	**36.7**	**49**	65.5	**38**
Recall@100									
BM25	92.5	34.8	24.6	34.4	16.7	55.9	74.4	71.8	50.6
SPECTER2	94.6	**45.7**	25	34.4	**26.8**	53.1	72	60.3	51.5
E5-base-v2	95.3	42.8	31.8	35.1	22	59.7	76.4	70	54.1
DyVo	93.4	36.3	28.5	15.4	20.9	53.1	75.9	66.8	48.8
ColBERT	88	34.2	27	33.9	17.5	51.9	72.5	63.4	48.6
SPLADEv2	89.3	35	26.8	35.2	17.8	54.4	75.8	66.9	50.2
$\text{ColBERT}_{\text{FRIEREN}}$	95.6	41.5	29.5	42.2	24.3	60.8	79.2	74	55.9
$\text{SPLADEv2}_{\text{FRIEREN}}$	95.7	42.5	27.2	39.3	23.2	62.7	78.6	72.4	55.2
$\text{SPLADEv2}^{\text{words}}_{\text{FRIEREN}}$	95.3	41.7	30.4	40	20.7	63.6	79.1	73	55.5
CASPER	95.8	43.5	**31.9**	41.6	25.2	65.6	82.1	72.8	57.3
CASPER++	**96.1**	43	**31.9**	**47.3**	23.1	**66.2**	**82.4**	**75.1**	**58.1**

Table 2: Retrieval performance on eight benchmark datasets. We report performance using nDCG@10 and Recall@100 (in percentage points). The best and second best results are **bolded** and underlined, respectively. We also provide Recall@1000 performance in Table 4.

cited elsewhere in the same paper but outside that specific citation group.

Citation contexts. The sentence in which a paper is cited often summarizes its key contribution as perceived by other authors. For each full-text document within the S2ORC corpus, we extract the citing sentences treat them as queries q_i. The cited paper's title and abstract form the positive document d_i^+, while a hard negative d_i^- is another paper cited elsewhere in the same document.

Author-assigned keyphrases. Keyphrases reflect an author's own view of their work's core concepts. We utilize two keyphrase generation datasets namely KP20K (Meng et al., 2017) and KPBioMed (Houbre et al., 2022), where documents are annotated with author-assigned keyphrases. For each entry, the comma-separated list of keyphrases serves as q_i, the paper itself as the positive document d_i^+, and a randomly sampled document from the same dataset acts as the negative d_i^-. We note that while random sampling yields easy negatives, we adopt this approach due to the lack of reliable heuristics for identifying hard negatives for this data type.

Titles. A document's title is a concise, author-provided summary of its content. Again, using the S2ORC corpus, we treat each paper's title as q_i and its corresponding abstract as d_i^+. The negative document d_i^- is the abstract of a paper cited within the full text of the source document.

6 Experiments

6.1 Datasets & Evaluation Metrics

We utilize a total of eight scientific retrieval datasets. Among them, three are sourced from the BEIR benchmarks (Thakur et al., 2021), namely SciFact (Wadden et al., 2020), SCIDOCS (Cohan et al., 2020) and NFCorpus (Boteva et al., 2016). The other five include DORIS-MAE (Wang et al., 2023), CSFCube (Mysore et al., 2021b), ACM-CR (Boudin, 2021), LitSearch (Ajith et al., 2024) and RELISH (Brown and Zhou, 2019). Table 3 summarizes the statistics of the evaluation datasets used in our experiments. To measure retrieval effectiveness, we employ nDCG@10 and Recall@100. We select nDCG@10 as the standard metric for ranking quality, while Recall@100 is included to evaluate the model's suitability as a first-stage retriever, as rerankers typically process the top-100 documents (Thakur et al., 2021). Unless noted, results represent a single run.

6.2 Retrieval Performance Evaluation

Baselines. To ensure a fair comparison, we select models with comparable parameter counts namely SPECTERv2 (Singh et al., 2023), E5-base-v2 (Wang et al., 2022), DyVo (Nguyen et al., 2024), SPLADEv2-max (Formal et al., 2021a), and ColBERT (Khattab and Zaharia, 2020), along with BM25. Additionally, we introduce CASPER++, a

simple ensemble of CASPER and BM25 obtained by summing their scores, following the approach of (Formal et al., 2022).

We also include versions of SPLADEv2-max and ColBERT finetuned on the FRIEREN dataset, denoted as $\text{SPLADEv2}_{\text{FRIEREN}}$ and $\text{ColBERT}_{\text{FRIEREN}}$, respectively. This setup serves two main purposes: 1) to assess the contribution of FRIEREN by comparing these models to their original versions; and 2) to ensure CASPER is evaluated against baselines trained under identical conditions. For both models, we employ similar hyperparameters as in the original papers, modifying only the training batch size to 20 and maximum sequence length to 256 to match CASPER's configurations. We provide further details regarding implementations of baseline methods in §B.2.

Comparison with baselines. The performance of our proposed method and the baselines are presented in Table 2. When compared to baselines not trained with FRIEREN, CASPER achieves better performance in terms of both nDCG@10 and Recall@100. Notably, CASPER surpasses E5-base-v2, despite the latter being trained on a much larger dataset.

CASPER demonstrates good performance when compared to baselines trained with FRIEREN. In terms of nDCG@10, CASPER outperforms $\text{SPLADEv2}_{\text{FRIEREN}}$, but slightly falls behind $\text{ColBERTv2}_{\text{FRIEREN}}$. In terms of Recall@100, CASPER outperforms both models. Overall, this demonstrates that even when trained on identical data, CASPER surpasses other strong baselines, indicating that the concept-integrated representation we introduce provides a tangible advantage.

To verify that performance gains are driven by the concept-integrated representation rather than just the expanded vocabulary, we conducted an additional comparison. In particular, CASPER was evaluated against $\text{SPLADEv2}^{\text{words}}_{\text{FRIEREN}}$, a version of $\text{SPLADEv2}_{\text{FRIEREN}}$ that included an extra vocabulary of 30k most common English words in $\mathcal{D}$ but not present in BERT's token vocabulary. This enhanced SPLADEv2 underperforms relative to CASPER, which further supports the effectiveness of the proposed concept-integrated representation.

FRIEREN generates suitable training data for scientific document search. Comparing baselines trained with FRIEREN to their original counterparts, we find that models trained with FRIEREN generally achieve competitive nDCG@10 and higher Recall@100. Overall, this shows that training data generated by FRIEREN is indeed beneficial for training scientific text retrievers.

Effectiveness of CASPER++. Finally, examining the performance of CASPER++, we observe that it surpasses CASPER, and also all evaluated baselines. This improvement demonstrates that integrating BM25 continues to provide complementary benefits to CASPER, which is consistent with the observation made in (Formal et al., 2022). Notably, for this integration, we tokenize inputs using CASPER's tokenizer, and therefore allow BM25 to operate on sequences of both tokens and keyphrases.

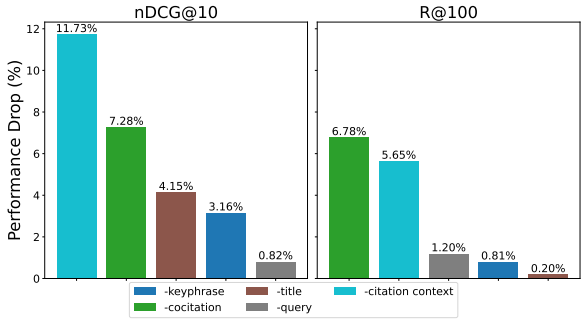

Figure 3: Retrieval performance drop as different data sources are removed.

6.3 Keyphrase Generation Performance

We evaluate CASPER on five standard benchmarks using the KPEval framework (Wu et al., 2023). Compared to CopyRNN (Meng et al., 2017), a strong seq2seq baseline, CASPER falls slightly behind in semantic F1. However, it generates more diverse keyphrases and is more than three times faster. A full experimental setup, quantitative results, and a qualitative case study comparing the two models are detailed in Appendix A.

6.4 Impact of Data Sources

In this section, we aim to understand the contribution of each data source within the FRIEREN framework to CASPER's overall performance. Specifically, we train five versions of CASPER, each by removing one type of data from the full training set. Results are presented in Figure 3, which shows that all data sources contribute positively to the final model, as omitting any single type leads to a decrease in performance.

Among the data types, *user queries* is the data type that results in insignificant performance drops across both evaluation metrics, indicating that the

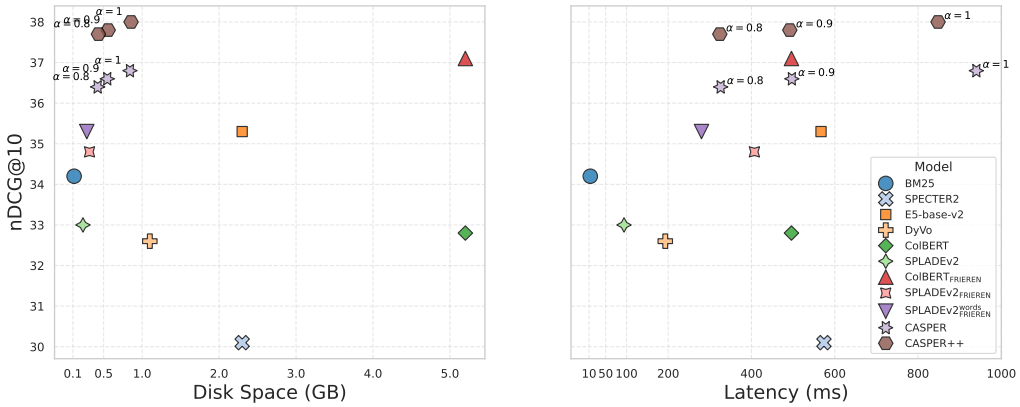

Figure 4: Trade-off between effectiveness (nDCG@10) and efficiency (Disk Space and Latency) for CASPER, CASPER++ (at varying pruning levels), and baselines. nDCG@10 is averaged across eight benchmark datasets. Efficiency metrics (Disk Space in GB; Latency in ms) are measured on CSFCube, the largest dataset, using a single-threaded CPU. Latency values exclude encoding time and are averaged over three runs.

contribution of user queries is limited. This observation further supports our argument regarding user queries and interaction data being an inadequate source of supervision for training retrievers in the scientific domain. In contrast, citation contexts and co-citations emerge as the most important sources of information, since removing either results in a substantial reduction in performance. Notably, both citation contexts and co-citations are readily available and can be scaled up with ease, unlike user queries, which are more challenging to collect and contribute less to the final outcome.

6.5 Efficiency

We investigate the impact of pruning CASPER's query and document representations on both effectiveness and efficiency. To prune a sparse vector w, we retain its α-mass subvector, following the definition proposed by Bruch et al. (2024). We conduct experiments with $\alpha \in \{0.8, 0.9, 1.0\}$, where $\alpha = 1.0$ represents the default CASPER.

Figure 4 illustrates the trade-off between retrieval effectiveness and efficiency across varying pruning levels, alongside competitive baselines. Regarding storage efficiency, we observe that sparse modeling baselines are superior to their dense counterparts, requiring significantly less disk space. In terms of latency, most models are comparable, clustering within the 200 to 600ms range. While the default CASPER ($\alpha = 1.0$) exhibits higher latency, the pruned variants ($\alpha \in \{0.8, 0.9\}$) achieve competitive inference speeds without compromising retrieval performance.

6.6 Ablation Studies

We summarize our key ablation studies in this section. First, in §D.1, we examine the impact of the concept-level score by varying β (see §4.4); our results indicate that concept-based matching yields optimal performance when it complements, rather than dominates, token-based matching. Second, in §D.3, we analyze the effect of keyphrase vocabulary variations. We find that an intermediate vocabulary size is most effective; and that ensuring comprehensiveness during vocabulary construction (see §4.1) is necessary. Finally, supporting our discussion in §4.2, we empirically demonstrate in §D.2 that sum pooling outperforms max pooling for generating concept representations.

7 Conclusion

In this paper, we present CASPER, a sparse model for scientific document retrieval designed to represent queries and documents by their research concepts and match them at both granular and conceptual levels. We also propose FRIEREN, a framework that mines data from scholarly references to overcome the lack of suitable supervision signals for scientific document search. Through extensive experiments, we show that CASPER generally outperforms strong baselines across eight scientific retrieval benchmarks. We also explore the effectiveness-efficiency trade-off via representation pruning and demonstrate CASPER's interpretability by showing its utility in the task of keyphrase generation.

Limitations

We acknowledge several limitations in our current work. **First**, we did not incorporate knowledge distillation techniques, as explored in prior studies (Formal et al., 2021a; Santhanam et al., 2022b). Applying distillation from strong cross-encoders or LLMs could likely further enhance the performance of CASPER and other baselines trained on FRIEREN. **Second**, computational constraints limited the scale of our experiments; although FRIEREN is capable of generating extensive training data, we have not yet explored the effects of scaling up to larger models and datasets. We leave these investigations to future work.

Finally, CASPER operates on a fixed keyphrase vocabulary, requiring retraining to incorporate new or modified concepts. In contrast, approaches like DyVo (Nguyen et al., 2024) offer greater flexibility by retrieving entities and scoring them based on their descriptions. However, DyVo relies on a frozen retriever and lacks end-to-end optimization. A promising direction for future research is to bridge this gap, combining CASPER's effective end-to-end learning with the dynamic flexibility of description-based models.

References

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*.

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

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 716–722. Springer.

Florian Boudin. 2021. Acm-cr: a manually annotated test collection for citation recommendation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 280–281. IEEE.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1126, Online. Association for Computational Linguistics.

Wichor M Bramer, Gerdien B De Jonge, Melissa L Rethlefsen, Frans Mast, and Jos Kleijnen. 2018. A systematic approach to searching: an efficient and complete method to develop literature searches. *Journal of the Medical Library Association: JMLA*, 106(4):531.

Peter Brown and Yaoqi Zhou. 2019. Large expert-curated database for benchmarking document similarity detection in biomedical literature search. *Database*, 2019:baz085.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient inverted indexes for approximate retrieval over learned sparse representations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 152–162.

Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. 2018. Solvent: A mixed initiative system for finding analogies between research papers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–21.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lam Do, Pritom Saha Akash, and Kevin Chen-Chuan Chang. 2023. Unsupervised open-domain keyphrase generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10614–10627, Toronto, Canada. Association for Computational Linguistics.

Lam Thanh Do, Aaditya Bodke, Pritom Saha Akash, and Kevin Chen-Chuan Chang. 2025. Eru-kg: Efficient reference-aligned unsupervised keyphrase generation. *arXiv preprint arXiv:2505.24219*.

Jeffrey M Dudek, Weize Kong, Cheng Li, Mingyang Zhang, and Michael Bendersky. 2023. Learning sparse lexical representations over specified vocabularies for retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3865–3869.

Khalid El-Arini and Carlos Guestrin. 2011. Beyond keyword search: discovering relevant scientific literature. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 439–447.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2353–2359.

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

Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, and 1 others. 2018. Science of science. *Science*, 359(6379):eaao0185.

Michael Gusenbauer and Neal R Haddaway. 2021. What every researcher should know about searching–clarified concepts, search advice, and an agenda to improve finding in academia. *Research synthesis methods*, 12(2):136–147.

Dorit S Hochbaum. 1997. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. *Approximation algorithms for NP-hard problems*, pages 94–143.

Maël Houbre, Florian Boudin, and Beatrice Daille. 2022. A large-scale dataset for biomedical keyphrase generation. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 47–53, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sebastian Huettemann, Roland M Mueller, and Barbara Dinter. 2025. Designing ontology-based search systems for research articles. *International Journal of Information Management*, 83:102901.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.

Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C Wallace. 2018. Learning disentangled representations of texts with application to biomedical abstracts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 4683.

SeongKu Kang, Bowen Jin, Wonbin Kweon, Yu Zhang, Dongha Lee, Jiawei Han, and Hwanjo Yu. 2025. Improving scientific document retrieval with concept coverage-based query set generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 895–904.

SeongKu Kang, Yunyi Zhang, Pengcheng Jiang, Dongha Lee, Jiawei Han, and Hwanjo Yu. 2024. Taxonomy-guided semantic indexing for academic paper search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7169–7184, Miami, Florida, USA. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2013. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742.

Yuta Kobayashi, Masashi Shimbo, and Yuji Matsumoto. 2018. Citation recommendation using distributed representation of discourse facets in scientific articles. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 243–251.

Mikalai Krapivin, Aliaksandr Autaeu, Maurizio Marchese, and 1 others. 2009. Large dataset for keyphrases extraction.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2021a. Multi-vector models with textual guidance for fine-grained scientific document similarity. *arXiv preprint arXiv:2111.08366*.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 4453–4470, Seattle, United States. Association for Computational Linguistics.

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021b. Csfcube–a test collection of computer science research articles for faceted query by example. *arXiv preprint arXiv:2103.12906*.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

Mariana Neves, Daniel Butzke, and Barbara Grune. 2019. Evaluation of scientific elements for text similarity in biomedical publications. In *Proceedings of the 6th Workshop on Argument Mining*, pages 124–135.

Thong Nguyen, Shubham Chatterjee, Sean MacAvaney, Iain Mackie, Jeff Dalton, and Andrew Yates. 2024. Dyvo: Dynamic vocabularies for learned sparse retrieval with entities. *arXiv preprint arXiv:2410.07722*.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *International conference on Asian digital libraries*, pages 317–326. Springer.

Biswajit Paria, Chih-Kuan Yeh, Ian EH Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing flops to learn efficient sparse representations. *arXiv preprint arXiv:2004.05665*.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

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)*, pages 7534–7550, Online. Association for Computational Linguistics.

Jianyou Andre Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. 2023. Scientific document retrieval using multi-level aspect-based queries. *Advances in Neural Information Processing Systems*, 36:38404–38419.

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. *arXiv preprint arXiv:2212.03533*.

Di Wu, Da Yin, and Kai-Wei Chang. 2023. Kpeval: Towards fine-grained semantic-based keyphrase evaluation. *arXiv preprint arXiv:2303.15422*.

Huanqin Wu, Baijiaxin Ma, Wei Liu, Tao Chen, and Dan Nie. 2022. Fast and constrained absent keyphrase generation by prompt-based learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11495–11503.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2020. Sparta: Efficient open-domain question answering via sparse transformer matching retrieval. *arXiv preprint arXiv:2009.13013*.

A CASPER for Keyphrase Generation

In this section, we show that through simple post-processing, CASPER can be used for the keyphrase generation task, which aims to identify present and absent keyphrases given a text d. Keyphrases generated by CASPER can be used to interpret its representations.

Present keyphrases. This type of keyphrases can be found within the input text. To identify present keyphrases, we first extract noun phrases from d to form a candidate set $C = \{c_i\}_i$. Next, we obtain sparse representation of d produced by CASPER $w^d = \{w_j^d\}_{j \in V_t \cup V_k}$. Then, highest-ranked candidates are chosen to be keyphrases, where the

ranking function is defined as

$$R(c_i) = \frac{\omega_d(c_i)}{|c_i| - \gamma} \sum_{j \in c_i} w_j^d \tag{8}$$

$$\omega_d(c_i) = 1 + \frac{1}{\log_2(\mathcal{P}_d(c_i) + 2)} \tag{9}$$

Here, γ and $\omega_d(c_i)$ are inspired by (Do et al., 2025). The former is a hyperparameter that controls the preference towards longer candidates, while the latter is applied to favor candidates that appear earlier in the text. $\mathcal{P}_d(c_i)$ is the offset position, computed by the number of words that precedes c_i in d. We note that the candidate c_i is tokenized over the combined vocabulary $V_t \cup V_k$. For example, "unsupervised machine learning", can be tokenized into ("un", "##su", "##per", "##vis", "##ed", "machine learning").

Absent keyphrases. Simultaneously, absent keyphrases are identified by selecting the keyphrases from V_k that are not in the document but possess the highest activation weights w_k^d in the sparse vector. A key advantage of this method is its exceptional speed. Specifically, by not relying on sequence-to-sequence generation (Meng et al., 2017) or external phrase retrieval for absent keyphrases (Do et al., 2025), CASPER offers a time-efficient generation capability.

B Implementation Details

B.1 CASPER

We initialize CASPER using DistilBERT$_{\text{base}}$. Unless otherwise specified, the keyphrase vocabulary size is set to $|V_k| = 30$k. After incorporating keyphrases into the base model (see §4.1), we conduct continual pretraining with a batch size of 64, a learning rate of 2×10^{-5}, and a total of 70k steps. To prioritize effective learning of new keyphrase tokens, we modify the masking strategy: within each sequence, 85% of keyphrase tokens are selected for prediction, while the rest of the masking procedure follows the original BERT methodology.

Following pretraining, we finetune CASPER using training data generated by FRIEREN (§5). To generate the dataset, we use only the first 7% of the S2ORC corpus[5] (20 files out of a total of 298 files). We sample a maximum of 1.5 million triplets per data type to manage resource usage and prevent over-representation of larger categories. As some types are smaller (e.g., *title* with 330,000 triplets), the resulting dataset yields 3.6 million triplets.

Hyperparameters for finetuning largely follows the settings described in (Formal et al., 2021b), with some adjustments. Specifically, we use a batch size of 20 due to resource limitations. For all experiments, we set $\lambda_q = 3 \times 10^{-4}$ and $\lambda_d = 1 \times 10^{-4}$ (see Eq. 2), while for the ranking loss, we set $\lambda_t = 1$ and $\lambda_k = 2$ (see Eq. 6). During inference, unless otherwise specified, we use $\beta = 0.25$.

B.2 Baselines Implementation Details

We utilize the public pretrained models of SPECTERv2[6], E5-base-v2[7] and SPLADEv2-max[8]. Our setup differs only in that we initialize with DistilBERT-base, use a batch size of 20, and train on MS MARCO using BM25 hard negatives, consistent with the SPLADE training protocol (Formal et al., 2021b,a). During inference, we employ the PLAID engine (Santhanam et al., 2022a) for both ColBERT and ColBERT$_{\text{FRIEREN}}$. Finally, for DyVo (Nguyen et al., 2024), we train the model using the official repository and report results for the DyVo-BM25 variant.

B.3 Processing CSFCube

For the CSFCube dataset, which provides relevance judgments for three aspects ("background," "method," and "result"), we combine these judgments into a single dataset (with 127, 73, and 81 judgments for each aspect, respectively). Queries with relevance judgments for multiple aspects are treated as distinct queries.

B.4 Computational Resources

We conduct all our experiments on a server with NVIDIA Ampere A40 GPUs (300W, 48GB VRAM each), along with two AMD EPYC 7302 3GHz CPUs and 256 GB of RAM. We employ 1 GPU at a time for all experiments.

C Keyphrase Generation Evaluation

To evaluate CASPER's keyphrase generation performance, we utilize five widely used keyphrase generation datasets: SemEval (Kim et al., 2013), Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin et al., 2009), and KP20K (Meng et al., 2017). For evaluation metrics, we adopt semantic F1 (SemF1), semantic recall (SemRecall),

[5] https://github.com/allenai/s2orc

[6] https://huggingface.co/allenai/specter2_base

[7] https://huggingface.co/intfloat/e5-base-v2

[8] https://huggingface.co/naver/splade_v2_max

Information Retrieval			
Dataset name	#Query	#Corpus	Avg D /Q
SciFact	300	5,183	1.1
SCIDOCS	1,000	25,657	4.9
NFCorpus	323	3,633	38.2
DORIS-MAE	100	363,133	16.4
CSFCube	50	776,070	5.64
ACM-CR	552	114,882	1.8
LitSearch	597	64,183	1.1
RELISH	1,684	163,170	80
Keyphrase Generation			
Dataset name	#doc	#kps/doc	%absent
SemEval	100	15.2	59.7
Inspec	500	9.8	22
NUS	211	11.6	49.3
Krapivin	460	5.7	51.2
KP20K	19,987	5.3	44.7

Table 3: Statistics of evaluation datasets

and diversity (emb_sim), as provided by the KPEval evaluation framework (Wu et al., 2023). In our evaluation, predictions are generated by selecting the top 5 present and top 5 absent keyphrases for each instance, resulting in a set of 10 predicted keyphrases per document.

Table 5 presents CASPER's performance on the keyphrase generation task. For comparison, we evaluate against CopyRNN (Meng et al., 2017), a well-established seq2seq keyphrase generation model, using the implementation from (Do et al., 2023). CASPER attains 86% of CopyRNN's semantic F1 score, reflecting weaker agreement with ground-truth keyphrases. However, CASPER achieves 94% of CopyRNN's semantic recall and generates substantially more diverse keyphrases, suggesting that precision is the more major contribution of CopyRNN's higher F1, which may be partly due to generating repetitive or highly similar phrases. Although not being able to outperform CopyRNN in terms of groundtruth agreement, CASPER offers significant practical advantages, as it is more than three times faster and generates more diverse keyphrases.

To supplement the automatic metrics, Table 6 presents the keyphrases generated by each model for a sample document. Both systems generate reasonable present keyphrases, but they differ when it comes to absent keyphrases. CopyRNN mainly forms absent phrases by recombining existing terms, whereas CASPER proposes truly novel phrases, one of which ("computer vision") matches the ground-truth annotation. In addition, CopyRNN tends to generate keyphrases that are similar ("nearest neighbor search" and "nearest neighbor"; "mobile robot vision" and "robot vision"), which supports our analysis above.

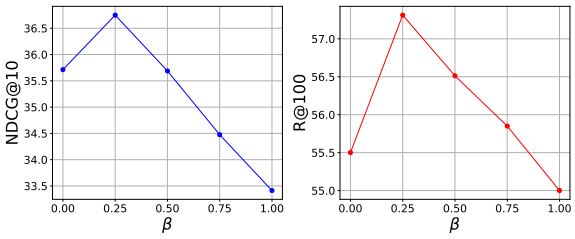

Figure 5: Retrieval performance (averaged across eight datasets) with different values of β

D Ablation Studies

D.1 Influence of Concept-based Representation

As described in §4.4, the final ranking score in CASPER is computed as a combination of token-level and keyphrase-level (concept-level) scores. To assess the impact of the concept-level score on overall performance, we vary the weighting parameter β (see Eq. 7) across five values: $\{0, 0.25, 0.5, 0.75, 1\}$.

The results, illustrated in Figure 5, reveal that CASPER achieves its highest performance at $\beta = 0.25$. Notably, when concept-based matching is not used ($\beta = 0$), the performance of CASPER drops significantly. Conversely, a β with value beyond 0.25 also results in a continuous decline in performance as it is increased. These findings indicate that, while concept-based matching provides clear benefits, it should complement rather than dominate token-based matching for optimal results.

D.2 Sum vs Max Pooling

To support our discussion in §4.2 on the effect of apply sum versus max pooling for forming concept-based representation, we train a version of CASPER where max pooling is used instead of sum pooling. The results, presented in Table 7, show that the default version which uses sum pooling achieves better performance.

D.3 Ablation Studies of Keyphrase Vocabulary

It is important to understand the impact of different keyphrase vocabularies to the final outcome of the model. In this section, we investigate: 1) how

	SciFact	SCIDOCS	NFCorpus	DORIS-MAE	CSFCube	ACM-CR	LitSearch	RELISH	AVG
				Recall@1000					
BM25	97.7	56.4	37	77.1	46.4	71.5	90	93.4	71.2
SPECTER2	99.7	**77.8**	59.7	70.9	53.8	76.1	87.1	90	76.9
E5-base-v2	99.7	70	**64.7**	69.6	54.4	77.6	91.4	93.6	77.6
DyVo	99.0	59.3	52.4	38.5	44.6	72.1	90.6	90.6	68.4
ColBERT	96	57	45.3	65.9	44.8	70.4	87.6	88.2	69.4
SPLADEv2	97.7	57	52.2	65.6	43	73.4	92.8	91	71.6
ColBERTv2$_{\text{FRIEREN}}$	99.3	66.5	56.1	77	52.7	81.2	91.6	94.8	77.4
SPLADEv2$_{\text{FRIEREN}}$	99.3	68.4	57.7	76.9	54.6	79.6	93	94.8	78
SPLADEv2$^{\text{words}}_{\text{FRIEREN}}$	98.7	68.8	62.1	75.7	54.7	80.6	93.3	95.2	78.6
CASPER	**100**	70.6	63.1	77.1	**60.2**	81.3	**94.9**	95.4	80.3
CASPER++	**100**	69.3	63.5	**81.5**	57.9	**82.6**	94.8	**96**	**80.7**

Table 4: Recall@1000 of baselines on eight retrieval benchmarks. The best and second-best results are **bolded** and underlined.

	CopyRNN			CASPER		
	SemF1↑	SemR↑	Diversity (emb_sim↓)	SemF1↑	SemR↑	Diversity (emb_sim↓)
Inspec	55.0	52.5	23.8	52.6	53.7	13.0
NUS	57.8	55.0	28.3	49.3	50.9	13.1
Krapivin	54.2	57.2	24.3	42.2	49.7	10.9
SemEval	50.4	43.0	24.8	46.7	42.1	12.5
KP20K	54.1	57.2	26.0	43.9	51.7	11.9
AVG	54.3	53.0	25.4	47.0	49.6	12.3
Throughput↑	11 (doc/s)			36.3 (doc/s)		

Table 5: Keyphrase generation performance on five benchmark datasets, with SemF1, SemR, and emb_sim reported in percentage points. ↑ indicates higher is better, and ↓ indicates otherwise. Throughput, measured in documents per second (doc/s), is averaged over five runs

Document	Keyphrases
Title: Adaptive road detection via context-aware label transfer. **Abstract**: The vision ability is fundamentally important for a mobile robot. Many aspects have been investigated during the past few years, but there still remain questions to be answered. This work mainly focuses on the task of road detection, ...	**CopyRNN**: nearest neighbor search; mobile robot; label transfer; depth map; nearest neighbor; mobile robot vision; context-aware robot; road recovery computing; road recovery **CASPER**: depth map; adaptive road detection; road detection; context-aware label transfer; robot; intelligent vehicle; detection algorithm; computer vision; adaptivity; transfer process **Targets**: road detection; depth map; label transfer; context-aware; mrf; computer vision

Table 6: Keyphrases generated for an example document, by CopyRNN and CASPER. Highlighted in orange and blue are present and absent keyphrases, respectively

Sum		Max	
NDCG@10	R@100	NDCG@10	R@100
36.8	57.3	35.7	56.2

Table 7: Average retrieval performance across eight datasets when sum and max pooling are employed for concept-based representation

Variation	NDCG@10	R@100
Main model		
$\lvert V_k \rvert$ = 30k	36.8	57.3
Varying vocabulary size		
$\lvert V_k \rvert$ = 5k	36.2	56.3
$\lvert V_k \rvert$= 15k	37.1	57.2
$\lvert V_k \rvert$ = 60k	35.6	55.8
Frequency based keyphrase vocabulary		
Frequency-based keyphrase vocab	36.2	56.8

Table 8: Retrieval performance of CASPER with different keyphrase vocabularies. We report the average retrieval performance across eight datasets

varying the size of the keyphrase vocabulary affects effectiveness; and 2) whether explicitly ensuring comprehensiveness is essential in constructing the vocabulary, or if simply selecting the most frequent keyphrases suffices (since a vocabulary including the most frequent keyphrases is also comprehensive to a certain extent). To explore this, we train three additional versions of CASPER with $|V_k| =$ 5k, 15k and 60k. Furthermore, we train another version of CASPER whose vocabulary are formed by selecting the most frequent 30k keyphrases.

The results, presented in Table 8, show that CASPER achieves the best performance with the default vocabulary setting. Firstly, analyzing performance as we vary vocabulary size, intermediate vocabulary sizes ($|V_k| =$ 15k and 30k) yield better results than either very small ($|V_k| =$ 5k) or very large ($|V_k| =$ 60k) vocabularies. The results suggest that the best performance is achieved when the keyphrase vocabulary is both common and comprehensive. In particular, a vocabulary that is too small is not comprehensive and therefore may lack

Mode	#Training data	NDCG@10	R@100
single	600k	25.1	50.8
	1.5M	24.9	53.3
	2.9M	25.4	52.2
multi	600k	24.2	50
	1.5M	25.2	52.5
	3.6M	27	53.6

Table 9: CASPER performance in multi and single disciplinary setting. We report average performance across the four Computer Science retrieval benchmarks, namely DORIS-MAE, CSFCube, LitSearch and ACM-CR

descriptive power (as it does not allow representing key concepts of many documents). On the other hand, one that is too large introduces specific, low-frequency keyphrases, which not only make it harder to train the model but also offer limited benefit regarding the enhancement of the representation space, as they produce matches fewer documents.

Secondly, the variant using the 30k most frequent keyphrases exhibits similar recall to the default but lower nDCG. This result suggests that there are benefits in explicitly optimizing comprehensiveness when building keyphrase vocabulary.

D.4 Multi versus Single Disciplinary

In this section, we evaluate CASPER's performance under two scenarios: single-disciplinary and multi-disciplinary training. Specifically, we investigate the performance of CASPER when tailored to a single discipline (specifically Computer Science in this study) versus when it is trained across multiple scientific fields.

To test this, we create a Computer Science-specific version of CASPER following the same procedure as the default model, with two key differences: 1) the keyphrase vocabulary V_k is constructed using only the Computer Science subset of $\mathcal{D}$, and continuous pretraining is performed also on this subset; 2) using FRIEREN, we generate a Computer Science training dataset by retaining only triplets where the positive document belongs to the Computer Science domain[9]. This results in a training set of comparable size (2.9M) to the one used to train the default model (3.6M).

We evaluate both versions across three training data scales: 600k, 1.5M triplets, and full (3.6M for multi-disciplinary and 2.9M for single-disciplinary). For fair comparison, we conduct experiments on four retrieval benchmarks whose main theme is Computer Science, namely DORIS-MAE, CSFCube, LitSearch and ACM-CR. The results, presented in Table 9, reveal an interesting pattern. When trained on 600k triplets, the single-disciplinary CASPER consistently outperforms its multi-disciplinary counterpart. However, as training data increases, the multi-disciplinary version improves and eventually achieves slightly superior performance when using 3M triplets. This suggests that while domain-specific training provides advantages with limited data, the multi-disciplinary approach benefits more from increased training scale. We attribute this to cross-domain knowledge transfer, where knowledge from different domains complement one another to achieve better results.

[9]To determine if a document belongs to the Computer Science domain, we utilize the "fieldsOfStudy" field returned by Semantic Scholar API