SCIREPEVAL: A MULTI-FORMAT BENCHMARK FOR SCIENTIFIC DOCUMENT REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned representations of scientific documents can serve as valuable input features for downstream tasks, without the need for further fine-tuning. However, existing benchmarks for evaluating these representations fail to capture the diversity of relevant tasks. In response, we introduce SciRepEval, the first comprehensive benchmark for training and evaluating scientific document representations. It includes 25 challenging and realistic tasks across four formats: classification, regression, ranking and search. We then use the benchmark to study and improve the generalization ability of scientific document representation models. We show how state-of-the-art models struggle to generalize across task formats, and that simple multi-task training fails to improve them. However, a new approach that learns *multiple* embeddings per document, each tailored to a different task format, can improve performance. We experiment with task-format-specific control codes and adapters in a multi-task setting and find that they outperform the existing single-embedding state-of-the-art by up to 1.5 points absolute.

1 INTRODUCTION

Learning representations of whole documents is critical for a variety of NLP tasks including classification, search, and recommendation (Cohan et al., 2020). Recent work has shown how pretrained language models (e.g., (Devlin et al., 2019; Raffel et al., 2020a; Brown et al., 2020)) can be tailored to produce high-quality representations of documents by training on contrastive learning objectives (Xu et al., 2021; Gao et al., 2021; Neelakantan et al., 2022). In the scientific domain, training objectives based on contrastive learning of cross-document links (e.g., citations) have shown further improvements in document-level representation learning (Cohan et al., 2020; Ostendorff et al., 2022b; Mysore et al., 2022). These methods are especially useful because the representations they produce can be indexed and later efficiently consumed by lightweight downstream models without additional fine-tuning.

While there has been significant progress in evaluating generalizability of NLP models (Ye et al., 2021; Sanh et al., 2021), evaluation of scientific document representation models has remained limited. Existing benchmarks either focus on document similarity (Mysore et al., 2021; Voorhees et al., 2020) or include tasks that are highly correlated and not diverse (Cohan et al., 2020).

We introduce SciRepEval, the first benchmark for supporting comprehensive evaluation of document-representation learning models in the scientific domain. Unlike prior work, SciRepEval is large and includes a collection of highly diverse tasks, thus encouraging research on various forms of generalization (including instance-level, cross-task and cross-domain generalization). It consists of 25 realistic tasks that reflect practical use cases of scientific document representations across four formats: text classification, regression, proximity-based ranking (e.g., nearest-neighbor), and ad-hoc search. SciRepEval contributes a standard set of both training and evaluation datasets to simplify comparisons between methods evaluated on the benchmark.

We then use this new benchmark to investigate and improve the generalization ability of document representation models. Following recent work (Cohan et al., 2020; Ostendorff et al., 2022b; Mysore et al., 2022) we further pre-fine-tune a transformer language model (SciNCL; Ostendorff et al., 2022b) to produce high-quality representations for downstream tasks. We hypothesize that condensing all relevant information of the document into a single vector representation might not be expressive enough for generalization across a wide range of tasks. Prior work addresses a similar

challenge in the context of document similarity by learning multiple finer-grained representations, each associated with a different *aspect* of a paper (e.g., task, method, results, etc) (Mysore et al., 2022; Ostendorff et al., 2022a). In contrast, we aim to learn effective representations for multiple downstream task *formats*.

Following recent success in multi-task learning in NLP (Ye et al., 2021; Sanh et al., 2021), we explore large-scale multi-task training in the context of scientific document representations, where we apply specific optimization objectives suitable for the various task formats in SciRepEval. i.e., cross-entropy loss for classification, triplet loss for proximity/ad-hoc search, and mean square error loss for regression. We explore two state-of-the-art techniques for generating format-specific document representations: using control codes (Keskar et al., 2019; Raffel et al., 2020a) as input indicating the format, and parameter-efficient adapter methods (Houlsby et al., 2019; Pfeiffer et al., 2021b; Stickland & Murray, 2019), in which a separate network module is introduced for each task format.

Our experiments investigate: (*i*) if existing document representation methods have the ability to generalize to a highly diverse set of tasks, (*ii*) if multi-task training on diverse data can improve document representation models, and (*iii*) if task-format-specific representations can improve generalization. Through extensive analysis we find that existing state-of-the-art scientific document representation models such as SPECTER and SciNCL Cohan et al. (2020); Ostendorff et al. (2022b) struggle with generalizing to all task types. We interestingly find that simple multi-task training on large set of tasks is *not* able to significantly improve the results. However, we learn that multiple task format-specific representations can substantially improve generalization.

To summarize, our contributions are:

(i) SciRepEval, a new comprehensive benchmark of highly diverse tasks for scientific document representation techniques covering 25 practical tasks across four different formats.

(ii) An extensive investigation on the generalization ability of state-of-the-art scientific document representation models.

(iii) A set of new multi-task document representation models that, unlike existing methods, can produce representations tailored to different task formats. The new methods show improved generalization over previous work, outperforming prior methods by up to 1.5 points absolute.

2 BACKGROUND

Representing Scientific Documents Earlier work aimed at document embeddings used word vectors (J et al., 2016; Le & Mikolov, 2014; Wu et al., 2018), convolutions (Liu et al., 2017; Zamani et al., 2018), bi-encoder networks (Conneau et al., 2017) and BERT-based methods (Reimers & Gurevych, 2019). Recent works have produced large scale language models pre-trained on scientific corpora (Beltagy et al., 2019; Yasunaga et al., 2022; Walker et al., 2021). These tend to perform better than general purpose models on scientific domain tasks, and serve as a foundation for learning dense embeddings of scientific documents. Cohan et al. (2020) and Ostendorff et al. (2022b) fine-tune SciBERT (Beltagy et al., 2019) with a triplet loss that encourages papers citing each other to have similar embeddings, using the title and abstract of research papers as the input.

Both Cohan et al. (2020) and Ostendorff et al. (2022b) are evaluated on the SciDocs benchmark. However, 4 of the 7 tasks in SciDocs are correlated with the citation prediction training objective and further, contain only easy negative candidates. Hence, the existing techniques work reasonably well on SciDocs. In contrast, SciRepEval provides a more challenging and diverse set of tasks, for both training and evaluation to help motivate methods for producing scientific document representations that can do well across multiple task formats. As a first step in this direction, we attempt to learn task-specific embeddings of the documents by pre-fine-tuning on multiple objectives simultaneously. Related to our approach, there are techniques in learning multiple embeddings per paper (Ostendorff et al., 2022a; Mysore et al., 2022). These methods are, however, orthogonal to ours in that they generate an embedding per paper "facet", while we focus on learning separate embeddings per task format. In addition, these techniques focus only on finer-grained paper similarity, while our aim is producing general embeddings applicable to a variety of task formats.

Multi-Task Learning Across Formats Multi-task learning (Caruana, 1993) with deep neural networks has been shown to improve performance over single-task training for related objectives (Liu et al., 2015; 2019b). Though unrelated tasks can lead to negative transfer, recent work has shown

that simply increasing the number of tasks tends to yield better performance in multi-task learning (Aghajanyan et al., 2021; Aribandi et al., 2022; Padmakumar et al., 2022). Aghajanyan et al. (2021) pre-fine-tune pre-trained language models simulataneously on 46 tasks across 4 task types before fine-tuning on the downstream task. Aribandi et al. (2022) pre-train T5 (Raffel et al., 2020b) on a combination of C4 span denoising and 107 other tasks across 8 task families. Ye et al. (2021) introduce an ontology of 160 tasks for few shot multi-task training. Unlike these task families, which are divided primarily by semantics (e.g., classifying sentiment vs classifying entailment), the training tasks in SciRepEval consist of 8 large-scale scientific datasets across the four task *formats*. Since our goal is to evaluate final document representations, rather than fine-tune on individual downstream tasks like the above approaches, we follow SPECTER (Cohan et al., 2020) and directly apply the representations as features to the tasks.

Adapters for Multiple Tasks Adapters were introduced by Houlsby et al. (2019) for parameter efficient training of transformers (Vaswani et al., 2017). A small number of trainable task-specific parameters are added to each layer, while freezing the base encoder. To apply adapters to multi-task learning, Pfeiffer et al. (2021a) define a two-step process they call Fusion. First, individual adapter modules are trained for every task. The second step introduces task-specific fusion modules at each layer which attend to (i.e. fuse) all the previously pre-fine-tuned adapters, keeping them fixed. Similarly, Stickland & Murray (2019) introduced Projected Attention Layers (PALs) with adapters and self-attention modules for every task, but the entire network is trained simultaneously.

Control Codes Control codes can be defined as token(s) pre-pended to the input to serve as additional signals to the model. Keskar et al. (2019) use control codes as prompts to govern style, content, and task-specific behavior for conditional text generation. Tay et al. (2022) use control codes to switch between three de-noising modes during pre-training, and associate a downstream task with a particular mode during fine-tuning. Zhang et al. (2022) apply control codes in the context of dense retrieval to produce multiple representations covering different aspects of the same document, allowing them to match queries written from multiple perspectives. In contrast to this past work, we use control codes to indicate target task format for the embedding output by the model, and demonstrate how this is effective for producing paper embeddings across different formats.

3 SciRepEval

We introduce SciRepEval, a benchmark suite of 25 tasks across four formats for training and evaluating multi-task embeddings of scholarly papers. SciRepEval aims to enable comprehensive evaluation of paper embeddings by providing (1) a highly diverse set of tasks—spanning multiple task formats such as classification, regression, proximity and ad-hoc search—to challenge the general-purpose applicability of embeddings, (2) realistic tasks that reflect practical use cases of paper embeddings, and (3) a standard set of both training and evaluation datasets to simplify comparisons between methods evaluated on the benchmark.

The previous scholarly paper embedding benchmark is SciDocs (Cohan et al., 2020), which includes two classification tasks, four nearest neighbors tasks, and one recommendation task. SciRepEval includes SciDocs as a subset, but addresses several key limitations:

(i) The four nearest neighbor tasks in SciDocs are constructed to distinguish a related document from randomly selected negatives given a query document, which might be too easy and not representative of real tasks in scholarly information retrieval. SciRepEval has more realistic tasks such as search, author disambiguation, and paper-reviewer matching among others.

(ii) For the methods evaluated in Section 5, we found that the SciDocs recommendations task was noisy and had limited power to distinguish between different embeddings. The test set includes only 1000 clickthrough events, and the use of propensity weighting means that an even smaller number of examples dominate test set performance. While SciRepEval includes SciDocs as a subset, we exclude the recommendations task.

(iii) The tasks in SciDocs were constructed to be used *only* for evaluation, and have few-enough samples that training on SciDocs is impractical (see Table 1). In SciRepEval, eight of the largest tasks across the four formats are used for training, while the rest out-of-train tasks are reserved for evaluation. This enables the study of multi-task approaches, rather than relying solely on the citation signal. The training data in SciRepEval also has a large scale representation in multiple domains as discussed in Appendix D.

Table 1: Summary of the SciRepEval benchmark tasks across the four formats - classification (CLF), regression (RGN), proximity (PRX) and adhoc search (SRCH). The models in section 6 are first trained on the in-train tasks and then benchmarked on their held-out sets as well as the 17 test tasks. Information retrieval tasks have **Q** queries with **P** candidate pairs and the S2AND task has **X** clusters with **Y** author-paper pairs. **S**: Silver, **G**: Gold. SciDocs is evaluated as per Cohan et al. (2020).

Task Format	Name	Train + Dev	Test	Eval Metric
		In-Train		
CLF	MeSH Descriptors	2,328,179	258,687	Macro F1
	Fields of study (FoS)	676,524 **S**	471 **G**	Macro F1
RGN	Citation count	202,774	30,058	Kendall's $\mathcal{T}$
	Year of Publication	218,864	30,000	Kendall's $\mathcal{T}$
PRX	Same Author Detection	**Q:** 76,489 **P:** 673,170	**Q:** 13,585 **P:** 123,430	MAP
	Highly Influential Citations	**Q:** 65,982 **P:** 2,004,688	**Q:** 1,199 **P:** 54,255	MAP
	Citation Prediction Triplets	819,836	—	—
SRCH	Search	**Q:** 723,343 **P:** 7,233,430	**Q:** 2,585 **P:** 25,850	nDGC
		Out-of-Train		
CLF	Biomimicry	—	11,057	Binary F1
	DRSM	—	7,520 **S**; 955 **G**	Macro F1
RGN	Peer Review Score	—	10,210	Kendall's $\mathcal{T}$
	h-Index of Authors	—	8,438	Kendall's $\mathcal{T}$
	Tweet Mentions	—	25,655	Kendall's $\mathcal{T}$
PRX	S2AND	—	**X:** 68,968 **Y:** 10,942	B^3 F1
	Paper-Reviewer Matching	—	**Q:** 34 **P:** 1,729	P@5, P@10
	Feeds-1	—	**Q:** 423 **P:** 4,223	MAP
	Feeds-M	—	**Q:** 9025 **P:** 87,528	MAP
SRCH	Feeds Title	—	**Q:** 424 **P:** 4,233	MAP
	TREC-CoVID	—	**Q:** 50 **P:** 69,318	nDCG
		SciDocs		
CLF	MAG	—	23,540	Macro F1
	MeSH Diseases	—	25,003	Macro F1
PRX	Co-view	—	**Q:** 1,000 **P:** 29,978	MAP, nDCG
	Co-read	—	**Q:** 1,000 **P:** 29,977	MAP, nDCG
	Cite	—	**Q:** 1,000 **P:** 29,928	MAP, nDCG
	Co-cite	—	**Q:** 1,000 **P:** 29,949	MAP, nDCG

The tasks in SciRepEval are summarized in Table 1. They are a mixture of existing and new datasets. Datasets with at least 100,000 instances (triplets for proximity/ad-hoc search) are *in-train* datasets used for training and those with fewer are *out-of-train* datasets used only for evaluation. Although SciDocs tasks are used as out-of-training evaluation tasks, we report their performance in a separate category.

Next, we briefly describe each of the task formats and their component tasks. Full details are provided in Appendix A. Except for *Search*, all the tasks use paper embeddings created from a combination of paper title and abstract as the input. Search requires additional metadata (subsection 4.1) which is concatenated to the title and abstract before producing the paper representation.

Ad-Hoc Search In ad-hoc search tasks, we are given a textual query and the task is to rank a set of candidate papers by relatedness to the query. Ad-hoc search is a critical mechanism for paper discovery in practice, and we gather multiple real-world data sets for training and evaluation. One evaluation dataset comes from previous work, TREC-CoVID (Voorhees et al., 2020), a biomedical challenge task that ranks papers from CORD-19 Wang et al. (2020b) in response to textual search queries. Two other datasets are newly introduced in our work: a 'feeds' dataset taken from a scholarly paper recommendation system, where we treat the user-specified feed name as the topic query, and the goal is to rank the papers the user has annotated as relevant to the feed above those annotated as irrelevant. Finally, for training, we release a new large data set of more than 700,000 clickthrough events from a scholarly search engine which we term as the *Search* task.

To evaluate an embedding set on ad-hoc search, we rank candidate papers by increasing Euclidean distance between the query embedding and the candidate paper embeddings. `Pytrec_eval` (Van Gysel & de Rijke, 2018) is used to calculate the ranking metrics. Normalized Discounted

Cumulative Gain (nDCG) is used for Search and TREC-CoVID tasks as the true relevance score can be > 1. For the feeds tasks which have binary labels, we use Mean Average Precision (MAP).

Proximity Similar to ad-hoc search, proximity tasks involve ranking a set of candidate papers by their relatedness to a query, except the query in this case is not textual but instead a paper. Proximity-based tasks form a basis for paper-based retrieval and recommendation, and for estimating paper similarity for use in applications like author disambiguation. We include a total of eleven proximity-based tasks, including four evaluation tasks from SciDocs (predicting citations and co-citations, and predicting co-viewed or co-read papers), and two others from previous work: the S2AND author disambiguation task (Subramanian et al., 2021) with paper similarity features, and Paper-Reviewer Matching, where candidate reviewers are ranked by expert annotators based on the similarity of their papers to the query paper to be reviewed. The Paper-Reviewer Matching task combines three existing datasets (Mimno & McCallum, 2007; Liu et al., 2014; Zhao et al., 2022) which we describe in more detail in subsection A.2. We also introduce five new proximity tasks including two feeds evaluation tasks from the recommender discussed above, where one or multiple relevant papers serve as queries. For training, we include three large-scale datasets aimed at predicting same-authors, citations (via triplets) (Cohan et al., 2020), and influential citations, which we define as four or more citations of the same paper in the text of a single paper.

For evaluating embeddings in proximity tasks, we rank candidates by Euclidean embedding distance, with MAP as the evaluation metric except for S2AND, which uses B^3 F1 (Bagga & Baldwin, 1998), and Peer Review Matching, which uses precision@5 and @10.

Classification Paper classification, in which the input is a paper and the output is a topical category, is a foundational task for document organization and discovery. Apart from the two SciDocs classification tasks (MAG and MeSH Diseases), we take four additional classification tasks, including a binary task to predict whether a paper is relevant to biomimicry (Shyam et al., 2019), two biomedical classification tasks, namely DRSM from Burns (2022) and MeSH Descriptors classification (Lipscomb, 2000), and a new large-scale field of study (FoS) multi-label training set of more than 500K papers with silver FoS labels based on publication venue.

We evaluate embeddings on classification tasks by scoring their performance as features within linear support vector classifiers. Results for these tasks are evaluated using F1 score (which may be micro- or macro-F1 depending on the dataset, indicated in Table 1). To better understand how embeddings perform in data-scarce regimes, we also construct two few-shot versions each from both out-of-train classification datasets and the FoS dataset subset for which we have manually annotated gold labels.

Regression We also consider a set of regression tasks where the goal is to predict a continuous quantity for a given paper. For evaluation, we consider predicting three numeric attributes related to prominence or quality: Tweet Mentions (Jain & Singh, 2021), and the peer review rating and maximum h-index of authors for a collection of ICLR papers obtained from OpenReview[1](forming two new datasets). For training, we introduce two additional datasets of more than 200K examples each, predicting citation count and year of publication.

We evaluate embeddings on regression tasks by scoring their performance when used as features within linear support vector regression models. Results for these tasks are evaluated using the Kendall's τ rank correlation between the true and predicted labels.[2]

4 MULTI-FORMAT REPRESENTATION LEARNING

Typical approaches for learning document embeddings produce a single embedding for every task (Cohan et al., 2020; Ostendorff et al., 2022b). We hypothesize that a single embedding will be insufficient for generalizing across a diversity of downstream tasks when the embeddings are used as features in lightweight classifiers. At the other extreme, learning embeddings for each task separately does not allow generalization to new tasks and also incurs significant storage costs scaling

[1]https://api.openreview.net

[2]We found in our experiments that Pearson's ρ and Kendall's τ produced similar relative results between models. We did not use MSE because its values are unbounded and could skew the overall average across the datasets in the benchmark.

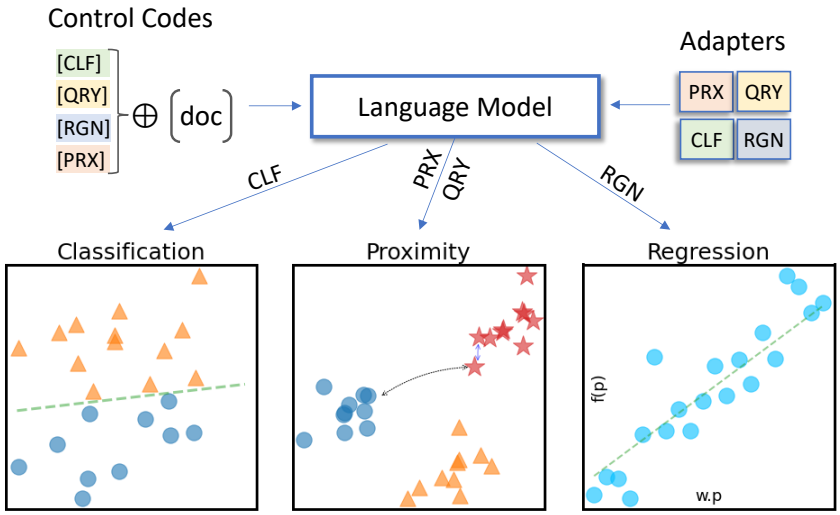

Figure 1: How multi-format embeddings are generated. A task format is either associated with a task-specific control code supplied with input document, or a task-specific adapter block attached to the model.

with the number of tasks. We propose method for learning a distinct document embedding for each task format, using a multi-task learning framework.

We assume we are given labeled data from a set of tasks for each of our four formats (ad-hoc search, proximity, classification, and regression), and we learn models capable of producing an embedding for any given (paper, format) pair. Here, papers are represented in terms of their title and abstract. Our goal is for the output embeddings to be useful in lightweight classifiers/regressors as well as for nearest neighbor tasks, which we evaluate both on held-out data from the training tasks, *and* on new held-out tasks.

To help build intuition for why different embedding sets for different task formats may be helpful, Figure 1 illustrates the qualitative distinctions between the task formats. In general, an embedding space performing well for one of the task formats may be less suited to the others; for example, the classifier embedding space in the figure provides an error-free linear classifier, but its nearest neighbor pairs are not always of the same class. Empirically, we find that learning specialized embeddings per format improves performance, and that embeddings trained on a format tend to perform better on held-out tasks with the same format (see Table 4). Further, partitioning randomly (as discussed in Section 7) was less effective than the format-based partitioning. Nonetheless, format-based partitioning is just one choice of many and experimenting with other partitioning schemes is an important item of future work.

4.1 Model

We follow Cohan et al. (2020) in using a pretrained transformer encoder as our base model. A scientific document is given as input to the encoder as a concatenation of its title and abstract separated by the [SEP] token [3]. Unlike Cohan et al. (2020), we use three different types of training objectives suitable for each format to train the model as described in subsection 4.2. We explore two methods to learn separate embeddings for each task form: control codes and adapters as shown in Figure 1.

Control Codes In the control code approach, we prepend a special per-format token (see Table 6 in the appendix) to the input and pass it to the transformer model, taking the final layer representation corresponding to this token as the document embedding and feeding it as input to the task-specific head (described in Section 4.2).

Adapters We also experiment with adapters which have been shown to be effective for multi-task learning. In particular, we explore Adapter Fusion (Pfeiffer et al., 2021a) and PALs (Stickland &

[3]For the Search task, additional metadata like paper venue and year of publishing is also supplied.

Murray, 2019) methods, each of which introduces task-specific adapters and attention modules at every transformer layer. Since our goal is to learn different embeddings for different task formats, we create modules for each task format rather than each task, and the final hidden representation of the [CLS] token output via the corresponding adapter is taken as the task format embedding of the document.

4.2 TRAINING

We train the model in a multi-task setup with task-heterogeneous batching (Aghajanyan et al., 2021). For classification and regression tasks, we use a linear head atop the base transformer encoder[4]. We train on both multi-class and multi-label tasks, using Cross Entropy loss for the former and Binary Cross Entropy (BCE) with sigmoid activation for the latter. For regression we minimize the Mean Square Error (MSE) loss.

For proximity and ad-hoc search tasks we use the triplet loss (similar to Cohan et al. (2020)). For these task forms, given a query, a relevance score accompanies each candidate. The query can be a document (for which we wish to find other similar documents) or a raw textual query. Each training instance in this setup is a triplet consisting of a paper or plain text query $\mathcal{Q}$, a positive candidate paper $\mathcal{P}+$ and a negative candidate $\mathcal{P}-$, where $\mathcal{P}+$ has a higher score than $\mathcal{P}-$. Then, we optimize the triplet loss:

$$L_{triplet} = \max\{d(\mathcal{Q}_E, \mathcal{P}_E^+) - d(\mathcal{Q}_E, \mathcal{P}_E^-) + \epsilon, 0\} \quad (1)$$

where d is the Euclidean distance used as a measure of similarity between the query embedding $\mathcal{Q}_E$ and candidate embeddings $\mathcal{P}_E^+$ and $\mathcal{P}_E^-$, and ϵ is the margin hyperparameter whose value is chosen as 1 based on preliminary experiments.

5 EXPERIMENT SETUP

Training Data We train our multi-format models on the 8 large scale in-train tasks detailed in Table 1. For the proximity and ad-hoc search tasks, we create up to 5 examples for each query by sampling positive and negative papers from its candidate pool. We limit the number of training samples from each task to at most 600K.[5] The resultant training and validation data sets consist of a total of 3.27M and 446K instances respectively.

Transformer Baselines As a first step, we evaluate the existing document representation methods on our benchmark. These include the transformer encoders SciBERT (Beltagy et al., 2019) – a language model pre-trained on scientific corpora; and SPECTER (Cohan et al., 2020), SciNCL (Ostendorff et al., 2022b) and ASPIRE (Mysore et al., 2022). ASPIRE produces representations for aspect-based matching between query and candidate papers which is a similar setting as our proximity tasks. Hence we only evaluate it on that specific subset and report the results in Appendix C. Next, for our multi-format baselines, we initialize with SciNCL which is the state of the art on SciDocs, and then further train it in a multi-task setup on the in-train tasks both with (MTL CTRL) and without the control codes (MTL CLS). Finally, to compare the control codes-based approach with the adapter techniques, we experiment with the BERT PALs and Fusion architectures, keeping SciNCL as the base model in both. Fusion being a two step process, first introduces task format specific adapters (Adapters) and then the fusion modules (Adapter Fusion). The MTL CTRL and adapter approaches produce multiple representations per document while MTL CLS produces a single representation similar to existing methods. We use the PyTorch implementations of the models by HuggingFace[6]. The specific training configurations are described in Appendix B.

6 RESULTS

Table 2 shows the evaluation of all our transformer baselines producing both single and multiple representations per document on SciRepEval. The pre-fine-tuned multi-format variants outperform the vanilla models on average, and we also find that all the approaches that produce multiple representation types outperform, by up to 1.5 points, the MTL CLS model, which learns only a single representation shared for all tasks. The adapter variants are better than MTL CTRL overall, and result in an improvement of 0.6-1.1 points on the out-of-train tasks with task-format specific adapters performing the best.

[4]The linear heads are thrown away after training.

[5]Performance with smaller dataset samples - max 400K samples/tasks was relatively poor.

[6]https://huggingface.co/models

Table 2: Evaluation results on SciRepEval in multiple settings. MTL CLS uses only a single embedding for all tasks, MTL CTRL and Adapter variants (Adapters, PALs, and Adapter Fusion) produce an embedding per task format using CTRL codes and embeddings respectively. We also consider an ensemble approach that averages the MTL CTRL and Adapter embeddings. The best results are highlighted in **bold**.

Model	In-Train	Out-of-Train	SciDocs	Average
SciBERT	51.2	52.6	69.0	58.1
SPECTER	54.3	57.4	89.1	67.9
SciNCL	55.3	57.8	**90.8**	69.0
SciNCL + MTL CLS	59.8	56.7	89.6	69.2
SciNCL + MTL CTRL	61.6	57.7	89.9	70.2
SciNCL + Adapters	61.4	**58.8**	90.3	70.7
SciNCL + PALs	61.9	58.3	90.0	70.5
SciNCL + Adapter Fusion	61.5	58.5	89.9	70.5
SciNCL + Adapters + MTL CTRL	**62.2**	**58.8**	90.7	**71.0**

Table 3: Results for multi-format training with SciBERT and SPECTER as base models. For brevity, we report only the single adapters results due to their additional advantage of computation efficiency. The best results for each base model are underlined.

Model	In-Train	Out-of-Train	SciDocs	Average
SciBERT + MTL CLS	59.0	57.2	88.9	69.0
SciBERT + MTL CTRL	61.9	57.9	89.5	70.2
SciBERT + Adapters	62.1	57.9	90.1	70.4
SciBERT + Adapters + MTL CTRL	62.6	58.5	90.4	70.9
SPECTER + MTL CLS	59.6	56.6	89.0	68.9
SPECTER + MTL CTRL	61.4	57.9	89.4	70.0
SPECTER + Adapters	61.1	58.6	89.6	70.3
SPECTER + Adapters + MTL CTRL	61.8	58.9	89.9	70.7

Further, as shown in Table 5, the control codes and adapters are the most efficient in terms of model size and computational efficiency. Hence, we try to improve upon each by combining representations from the Adapter model and the MTL CTRL model by averaging them[7], and we find that these combined embeddings outperform the individual ones consistently across the in-train, out-of-train, and SciDocs settings. All the models except SciBERT (not pre-trained with a citation objective) perform well on SciDocs, with vanilla SciNCL being the best, but the overall average on SciRepEval even with our best ensemble is 71.0, reaffirming SciRepEval's challenging nature. ASPIRE, as reported in Appendix C, performs well on SciDocs but not on other similar tasks in SciRepEval.

Alternative Base Models To confirm that our findings hold across multiple base models, we compare MTL CLS, MTL CTRL and adapters with SPECTER and SciBERT as the base models. Table 3 shows that the MTL CTRL token and the adapters approaches still substantially outperform the MTL CLS approach, suggesting that the efficacy of using an embedding per task format instead of a single embedding per document holds across a range of base model types.

7 Analyses

Specialization of Control Code Embeddings Our hypothesis is that by training embedding spaces on particular task formats, they will become more accurate for tasks of that format than for others. We test this hypothesis by sampling one in-train and one out-of-train[8] task of every format (for ease of computation) and applying *all* the control codes to them for evaluation. As shown

[7]We also tried concatenating the embeddings in preliminary experiments, which yielded similar results but doubled the embedding size.

[8]In-train: FoS, Citation Count, Same Author Detection, Search; Out-of-train: DRSM, Peer Review Score, Peer-Reviewer Matching, TREC-CoVID

Table 4: Cross task analysis for control codes. The best results for each task format across all control codes is underlined. These are represented in the diagonal for both in-train and out-of-train tasks suggesting that format based partitioning in multi-task training produces effective document representations suitable for the corresponding format.

Task format	Control Code Embeddings Used							
	Out-of-Train				*In-Train*			
	CLF	RGN	PRX	QRY	CLF	RGN	PRX	QRY
Classification	66.9	63.5	65.0	64.5	38.7	32.9	31.9	31.0
Regression	16.2	20.4	18.1	17.5	29.7	46.8	43.7	42.9
Proximity	43.8	43.1	44.7	44.7	87.1	82.2	89.0	88.3
Ad-hoc search	86.7	85.2	85.3	90.5	73.9	75.6	77.5	78.5

Table 5: Parameter and (relative) runtime efficiency comparison between models. MTL CTRL and vanilla Adapters are similar in runtime, but the PALs and Adapter Fusion variants add substantial computational cost.

Model	Parameters per Task Form	Training Time	Inference Time
MTL CTRL	768	1x	1x
PALs	2M	1.42x	1.29x
Adapters	1M	0.96x	1.05x
Adapter Fusion	22M	1.32x	1.69x

in Table 4, the control codes trained on a task format perform best for tasks of that format, for both in-train and out-of-train.

As an extension to this experiment we also analyze how well the control code representations work when the encoder is trained on tasks which are randomly grouped together as opposed to by task format. We take the mean evaluation metrics produced from 5 random partition runs. On the out-of-train tasks, the corresponding control codes for classification, regression, proximity and ad-hoc search show a gain of +0.2, +3.9, +4.5 and +2.2 points respectively over random partitioning. Similarly, for in-train tasks the control codes are better by +5.2, +3.8, +1.2 and +1.3 points respectively. The results suggest that representations specific to each task format do lead to better results overall.

Efficiency While the variants producing representations based on task-format serve as strong baselines on the SciRepEval benchmark as shown in Table 2, efficiency is another important consideration in practice. As shown in Table 5, the control code approach only requires one new control code embedding per format, and does not affect training time. PALs, by contrast, introduces new attention layers and trains the entire network, increasing training time, and Adapters adds and only trains half as many parameters as PALs. Fusion layers introduce 10x as many parameters as PALs leading to 2x more time on inference. Training and inference times are measured on runs with 1k and 10k samples, respectively.

8 Conclusion

We introduce SciRepEval, a benchmark for scientific document representation methods with 25 tasks across four task formats. On this benchmark, we show that learning a separate document representation for each task format substantially improves task performance compared to learning a single representation for all tasks. Exploring other partitioning schemes is one promising avenue for future work. In addition, we note that while our approach of producing one embedding per task format has substantially lower computation and storage costs than training a separate embedding for every task, it still requires more computation and storage than single-embedding approaches. Investigating ways to generate multiple embeddings in a single model pass or to dynamically choose the number of embeddings to generate could mitigate these costs.

REFERENCES

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5799–5811, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.468. URL https://aclanthology.org/2021.emnlp-main.468.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Vzh1BFUCiIX.

Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, 1998.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *EMNLP*, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Gully Burns. Drsm-corpus v1, 2022. URL https://github.com/chanzuckerberg/DRSM-corpus.

Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. Specter: Document-level representation learning using citation-informed transformers. *ArXiv*, abs/2004.07180, 2020.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070. URL https://aclanthology.org/D17-1070.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

Ganesh J, Manish Gupta, and Vasudeva Varma. Doc2sent2vec: A novel two-phase approach for learning document representation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pp. 809–812, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340694. doi: 10.1145/2911451.2914717. URL https://doi.org/10.1145/2911451.2914717.

Naman Jain and Mayank Kumar Singh. Tweetpap: A dataset to study the social media discourse of scientific papers. *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 328–329, 2021.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858, 2019.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196. JMLR.org, 2014.

Carolyn E. Lipscomb. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88 3:265–6, 2000.

Chundi Liu, Shunan Zhao, and Maksims Volkovs. Unsupervised document embedding with cnns. *arXiv: Computation and Language*, 2017.

Shengchao Liu, Yingyu Liang, and Anthony Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *AAAI*, 2019a.

Xiang Liu, Torsten Suel, and Nasir D. Memon. A robust model for paper reviewer assignment. In *RecSys '14*, 2014.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL*, 2015.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *ACL*, 2019b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *KDD '07*, 2007.

Sheshera Mysore, Timothy J. O'Gorman, Andrew McCallum, and Hamed Zamani. Csfcube - a test collection of computer science research articles for faceted query by example. *ArXiv*, abs/2103.12906, 2021.

Sheshera Mysore, Arman Cohan, and Tom Hope. 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*, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.331. URL https://aclanthology.org/2022.naacl-main.331.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pretraining. *ArXiv*, abs/2201.10005, 2022.

Malte Ostendorff, Till Blume, Terry Ruas, Bela Gipp, and Georg Rehm. Specialized document embeddings for aspect-based similarity of research papers. *2022 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 1–12, 2022a.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. Neighborhood contrastive learning for scientific document representations with citation embeddings. *ArXiv*, abs/2202.06671, 2022b.

Vishakh Padmakumar, Leonard Lausen, Miguel Ballesteros, Sheng Zha, He He, and George Karypis. Exploring the role of task transferability in large-scale multi-task learning. In *NAACL*, 2022.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *ArXiv*, abs/2005.00247, 2021a.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, Online, April 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020a. ISSN 1532-4435.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683, 2020b.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Vikram Shyam, Lauren Friend, Brian Whiteaker, Nicholas Bense, Jonathan Dowdall, Bishoy Boktor, Manju Johny, Isaias Reyes, Angeera Naser, Nikhitha Sakhamuri, Victoria Kravets, Alexandra Calvin, Kaylee Gabus, Delonte Goodman, Herbert Schilling, Calvin Robinson, Robert Omar Reid II, and Colleen Unsworth. Petal (periodic table of life) and physiomimetics. *Designs*, 3 (3), 2019. ISSN 2411-9660. doi: 10.3390/designs3030043. URL https://www.mdpi.com/2411-9660/3/3/43.

Asa Cooper Stickland and Iain Murray. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *ICML*, 2019.

Shivashankar Subramanian, Daniel King, Doug Downey, and Sergey Feldman. S2AND: A Benchmark and Evaluation System for Author Name Disambiguation. In *JCDL '21: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2021*, JCDL '21, New York, NY, USA, 2021. Association for Computing Machinery.

Yi Tay, Mostafa Dehghani, Vinh Quang Tran, Xavier García, Dara Bahri, Tal Schuster, Huaixiu Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *ArXiv*, abs/2205.05131, 2022.

Marco Valenzuela, Vu A. Ha, and Oren Etzioni. Identifying meaningful citations. In *AAAI Workshop: Scholarly Big Data*, 2015.

Christophe Van Gysel and Maarten de Rijke. Pytrec_eval: An extremely fast python interface to trec_eval. In *SIGIR*. ACM, 2018.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. Trec-covid: Constructing a pandemic information retrieval test collection. *ArXiv*, abs/2005.04474, 2020.

Nicholas Walker, Amalie Trewartha, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin Persson, Gerbrand Ceder, and Anubhav Jain. The impact of domain-specific pre-training on named entity recognition tasks in materials science. *Available at SSRN 3950755*, 2021.

Kuansan Wang, Iris Shen, Charles Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: when experts are not enough. *Quantitative Science Studies*, 1(1):396–413, February 2020a. doi: 10.1162/qss_a_00021. URL https://www.microsoft.com/en-us/research/publication/microsoft-academic-graph-when-experts-are-not-enough/.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Michael Kinney, Ziyang Liu, William Cooper Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D Wade, Kuansan Wang, Christopher Wilhelm, Boya Xie, Douglas A. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020b.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. Word mover's embedding: From Word2Vec to document embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4524–4534, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1482. URL https://aclanthology.org/D18-1482.

Peng Xu, Xinchi Chen, Xiaofei Ma, Zhiheng Huang, and Bing Xiang. Contrastive document representation learning with graph attention networks. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3874–3884, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.327. URL https://aclanthology.org/2021.findings-emnlp.327.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. In *ACL*, 2022.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *ArXiv*, abs/2104.08835, 2021.

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 497–506, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271800. URL https://doi.org/10.1145/3269206.3271800.

Shun Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. Multi-view document representation learning for open-domain dense retrieval. In *ACL*, 2022.

Yue Zhao, Ajay Anand, and Gaurav Sharma. Reviewer recommendations using document vector embeddings and a publisher database: Implementation and evaluation. *IEEE Access*, 10:21798–21811, 2022. doi: 10.1109/ACCESS.2022.3151640.

A SciRepEval Tasks

A.1 Ad-hoc search

Search We used clickthrough data from an academic search engine. Only search queries with at least 10 results were included, and a set of heuristic rules were applied to exclude likely noise and bots. We removed author queries when the query was judged to contain any person tokens by named entity recognition (Honnibal & Montani, 2017).

Feeds Research feeds help researchers maintain a library of papers they are currently reading and also index them by topics. We use anonymized research feeds data from an academic search engine that recommends papers based on a user's library. This data includes information on whether users found the recommendations relevant or not. The data contains 430 feeds which have more than five positive and two negative paper annotations from users. We use this to create the Feeds-1, Feeds-M and Feeds Title tasks. The first two are proximity tasks and are described in section 3.

Feeds Title: The title of a research feed provided by the user usually indicates the topic of scientific articles in the feed. While the other two datasets have papers as query and belong to the proximity family, this dataset is classified as ad-hoc search, as the query is a short text snippet rather than a paper. We remove feeds with generic titles like 'My field' and 'Final Project'; replace abbreviations with their long forms where possible and filter out feeds with non-English titles.

TREC-COVID TREC-COVID was introduced by Voorhees et al. (2020) as a biomedical literature search task relevant to COVID-19. The dataset consists of 50 search queries and candidate literature from the CORD-19 corpus Wang et al. (2020b) along with their relevance scores on a scale of 0-2. Each query consists of a short title, a question asking the required information and a narrative describes briefly exactly the type of information that the results should have. For our evaluation we combine these fields into a single text separated by the [SEP] token.

A.2 Proximity

S2AND and Same Author Detection The S2AND dataset (Subramanian et al., 2021) contains signatures (author-paper pairs) that are clustered according to which author mentions refer to the same person. Due to the high resource requirements of running the original S2AND evaluation, we create S2AND-mini, a version of S2AND with only 1000 blocks from each of S2AND's dataset sources and at most 500 signatures per block. Our evaluation of S2AND-mini follows the original evaluation of S2AND; that is, our method's document embeddings are used along with author and paper metadata to create features for a clustering algorithm that consists of a pairwise scoring model followed by greedy agglomerative clustering. We use B^3 F1 (Bagga & Baldwin, 1998) as in the original paper for evaluation.

We also use S2AND to create the data for our same-author detection task. Unlike the original S2AND evaluation, our same-author task uses only paper embeddings without any additional author or paper metadata, which allows us to directly train the embedding model on the data. Same-author detection is formulated as a triplet ranking task; given three papers of which two share an author, the goal is to find the matching pair.

Feeds-1 We re-purpose the feeds dataset from section A.1 for this and the next task. The first paper added to a feed chronologically serves as the query. The next 5 positive user annotations are considered relevant and 5 negative candidates are sampled either from user annotations or randomly.

Feeds-M Given K positive papers annotated in a feed (assuming $K > 5$), we use the first $M = K - 5$ as queries. For every query, the positive candidates are sampled from all the papers the user positively annotated after the query paper was added to their feed, and negative candidates are sampled from user annotations or randomly.

Peer Reviewer Matching In this task the goal is to judge whether a given paper is relevant to a potential reviewer. As data for this task is hard to obtain at scale due to the double-blind nature of many conferences and journals, we combine multiple existing reviewer-paper matching datasets:

- Mimno & McCallum (2007), with 393 paper-review relevance ratings from a corpus of 148 NeurIPS 2006 papers and 364 reviewers, annotated by nine human experts.
- Liu et al. (2014), an extension of Mimno & McCallum (2007) which adds 766 additional paper-review annotations.
- Zhao et al. (2022), with 694 paper-reviewer relevance ratings from a corpus of 75 papers and 1833 reviewers from the IEEE ICIP 2016 conference, annotated by 3 human experts.

All datasets have been annotated on the same 0-3 relevance rating scale. The candidate reviewers are all researchers, and we embed all the papers written by them using our models. To obtain the model's score for each candidate reviewer, we compute the cosine similarity between the query paper and each of the candidate's papers, and take the mean of the top 3 similarities as the score. We consider two ways to map the 0-3 relevance judgements to binary labels—hard and soft decision—where for the soft decision a score of 2 or 3 is considered relevant and for hard decision only a score of 3 is considered relevant. Precision at 5 (P@5) and 10 (P@10) results are used as the final metric, which ultimately results in four numbers (P@5 and P@10 for each of hard and soft decisions), which are averaged to produce the single number reported in our final results for this task.

Highly Influential Citations In this task, given a paper A and paper B, we aim to predict whether B is highly influenced by A. As measuring influence is subjective and human annotation is expensive, we approximate influence by counting the number of times A is cited in the text of B. If A is cited at least 4 times, we consider it to be highly influential (a positive example in our triplet-based loss); otherwise, we consider it to be a negative example. During evaluation, we sample query papers which have at least 5 positive candidates and compute the L2 distance for similarity ranking. Note that our definition of 'influential' differs from that in Valenzuela et al. (2015).

Citation Prediction (SPECTER Pre-training Triplets) This is the task and dataset used for pre-training in Cohan et al. (2020). It is based on citation links between scientific documents where each instance is a triplet consisting of a query, a positive and a negative paper. Each query can have up to five triplets, where the positives are sampled from papers directly cited by the query and negatives are chosen either randomly (easy) or from citations of citations (hard). 3 easy and 2 hard difficult are chosen for each query. To evaluate the effectiveness of this pre-training we follow Cohan et al. (2020) and use SciDocs for evaluation, excluding the recommendations task.

A.3 CLASSIFICATION

MeSH Descriptors Medical Subject Headings (MeSH) (Lipscomb, 2000) indexes biomedical publications into a categorical hierarchy consisting of descriptors which refer to topic headings and specific aspect related to a topic respectively. The dataset is a collection of scientific documents belonging to the 30 most frequently occurring top level MeSH descriptors and having exactly one qualifier. We filter out the records that don't have an associated qualifier. The descriptors thus serve as the labels in the multi-class classification task.

Fields of Study (FoS) The FoS task is a multi-label classification problem where each scientific document is assigned one or more classes out of 23 possible fields. For gold test data, we manually labeled 471 papers into at most three fields-of-study. For silver training data, we assumed that a paper within a venue generally falls within a narrow set of fields and manually assigned FoS labels to publication venues. We then propagated the venue labels to the papers published therein.

To evaluate different data sizes, we obtain the F1 score on the gold data in three settings: 5-shot, 10-shot, and the complete gold test set. The average of these scores is treated as the score for this task when computing the overall average score for the benchmark.

Disease Research State Model (DRSM) DRSM (Burns, 2022) is a collection of Pubmed papers that deal with six specific aspects of rare diseases. The gold data is annotated by in-house experts and used for evaluation, while the silver data is generated by annotation service providers with medical expertise.

Similar to FoS, we obtain the F1 score on 24-shot, 64-shot, and full data, then average the results before computing the final benchmark score.

Biomimicry We sample tags for a set of papers in the PeTaL database (Shyam et al., 2019) to create a binary classification dataset with labels indicating whether each paper is about biomimicry. The data is unbalanced, with only 13% positive samples. We evaluate 16-shot, 64-shot, and full-data setup and take the mean to get the final score.

A.4 REGRESSION

Citation Count We sample a collection of scientific articles published in 2016 from the set of papers in the search dataset described in A.1, so that a 5 year period has passed for them to collect citations. Each article has at least one citation, and the citation counts are converted to log scale.

Year of Publication The aim of this task is to determine research trends by predicting the year of publication of a scientific article. We sample publications from the search dataset with a publication date after the year 2005 and scale the years so that their values are between 0 and 1. Further, since this task is used for training along with citation count prediction, and to align the loss scales, the labels are scaled by the mean of the labels in citation count for parity.

Peer Review Score We use the OpenReview API[9] to collect paper metadata and corresponding review scores for ICLR conferences from 2017 to 2022. Each reviewer in ICLR assigns a final rating in the range [0-10], and we take the mean rating as the label for every paper.

h-Index of Authors In this task the goal is to predict the maximum h-Index of any of the authors of a scientific publication. We re-use the peer review score dataset, obtain the h-Index of all the authors for each paper using the Semantic Scholar API[10], and pick the max as the label. The labels are normalized to lie between [0,1].

Tweet Mentions The goal of this task is to predict the combined number of a paper's mentions and retweets. We post-process the dataset created by Jain & Singh (2021) containing tweets about Arxiv papers between 2010-19. The sum of normalized counts of mentions and retweets is finally considered as the score to be predicted.

B IMPLEMENTATION DETAILS

During pre-training, all the tasks with the same format share their task-format specific parameters. The control code based paradigm introduces four new (randomly-initialized) special tokens to the vocabulary. We try initializing these additional parameters randomly, with the [CLS] token and a combination of [CLS] with some noise. However, it has little impact on the resulting model performance with random initialization being better on average. Further, we also tried loss weighting strategies (Chen et al., 2018; Liu et al., 2019a) but our preliminary experiments produced better results without any scaling so we didn't explore it further. All the base models are trained for two epochs on two 48GB NVIDIA Quadro RTX 8000 GPUs with 16 bit precision, an effective batch size of 256, and a maximum input length of 512 tokens. Each batch is sampled with an equal number of examples from each task.[11] We use AdamW (Loshchilov & Hutter, 2019) with ϵ = 1e-8. The learning rate follows an inverse square root schedule with a linear warmup of 700 steps and peak of 5e-5.

The adapter approaches follow the two step training process and learning rate configurations described in Pfeiffer et al. (2021a). One adapter per task family is attached to the base model in both single adapter and fusion stages and is trained for a maximum of 6 and 4 epochs respectively. For PALs one layer is added per task format and the entire network is trained for 2 epochs as in Stickland & Murray (2019).

[9] https://api.openreview.net

[10] https://api.semanticscholar.org/

[11] We experimented with mixed and task sequential batching as well which did not yield good results.

Table 6: Assigned input formats and control codes for each task form. `[CLF]`, `[RGN]`, `[PRX]` and `[QRY]` are special tokens, `doc` is the input.

Task form	Input format
Classification	`concat([CLF],doc)`
Regression	`concat([RGN],doc)`
Proximity	`concat([PRX],doc)`
Ad-hoc Search	`concat([QRY]/[PRX],query/doc)`

B.1 EVALUATION

For classification and regression, we train a linear SVM on each downstream task using the embeddings as input, and we tune the regularization parameter $\mathcal{C}$ via grid search. Multi-class and multi-label classification are configured under the one vs all classifier setting.

C ASPIRE EVALUATION

Table 7: Comparison of the our SciNCL multi-format methods with ASPIRE on proximity tasks. The best results for each base model are underlined. TS: Text Supervision, OT: Optimal Transport

Model	In-Train	Out-of-Train	SciDocs	Average
TS ASPIRE$_{\text{CS}}$	65.0	65.2	91.3	79.9
TS ASPIRE$_{\text{Bio}}$	65.5	65.6	90.9	79.8
OT ASPIRE$_{\text{CS}}$	64.5	65.5	91.2	79.8
OT ASPIRE$_{\text{Bio}}$	65.0	65.7	90.5	79.6
SciNCL + MTL CTRL	66.9	66.9	91.1	80.7
SciNCL + Adapters	<u>67.0</u>	<u>67.5</u>	<u>91.5</u>	<u>81.1</u>

ASPIRE (Mysore et al., 2022) produces representations for the dense retrieval of scientific documents based on matching multiple aspects between the query and candidates. To evaluate these representations under the settings they are designed for, we only report the results on the proximity tasks in Table 7. We use the model implementations available on HuggingFace which have been pretrained on documents from the Computer Science (CS) and Biomedical (Bio) domains. The models variants can be further sub-categorized as retrieval based on best aspect matching (TS ASPIRE) and a weighted sum of the similarity score among all the aspects based on Optimal Transport (OT ASPIRE) between the query and candidates. Both our multi-format approaches with control codes and adapters produce better results overall and on out-of-train tasks. Note however, since ASPIRE models are trained on co-citations, they perform much better on average on the citation based tasks from SciDocs.

D SCIREPEVAL DOMAIN DISTRIBUTION

We study the domain diversity of SciRepEval and display the results in Table 8. To compare against the training data for SciDocs, we consider the citation prediction triplets on which SPECTER is trained which is also a subset of the SciRepEval in-train tasks. Even though Medicine and Computer Science papers still form a bulk of the data, SciRepEval has 105x more documents on average per domain compared to the SPECTER triplets.

E SPECTER OBJECTIVE

Lastly, we perform an ablation study to better understand the importance of the unsupervised citation-based training objective. We used SciBERT as the base model for this ablation since both SPECTER and SciNCL were trained with the citation objective. Removing the citation objective and its accompanying data from SciBERT + MTL CTRL, we find that the in-train performance drops

Table 8: Data domain distribution in SciRepEval for the training tasks and comparison with SciDocs. We group the unique documents in both the benchmarks by their MAG (Wang et al., 2020a) fields of study and present the counts in columns 2 and 3 and the absolute increase per field in column4.

Field of study	SciRepEval (A)	SciDocs (B)	Increase Ratio (A/B)
Medicine	3,201,323	74,685	43
Computer Science	1,187,689	199,664	6
Biology	882,357	13,377	66
Chemistry	508,056	3,813	133
Psychology	492,071	22,590	22
Materials Science	271,865	7,681	35
Engineering	254,826	31,444	8
Mathematics	231,482	25,800	9
Physics	217,670	7,285	30
Business	217,585	5,450	40
Sociology	156,128	2,305	68
Political Science	154,388	1,032	150
Economics	123,357	2,705	46
Environmental Science	91,682	1,136	81
Art	89,527	206	435
Geography	83,688	1,491	56
Philosophy	61,996	151	411
Geology	51,103	640	80
History	46,430	159	292

from 61.9 to 61.8, while out-of-train drops from 57.9 to 57.5, hinting that the citation objective may be helpful for generalization to new tasks.