Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings

Anonymous ACL submission

Abstract

Learning scientific document representations can be substantially improved through contrastive learning objectives, where the challenge lies in creating positive and negative training samples that encode the desired similarity semantics. Prior work relies on discrete citation relations to generate contrast samples. However, discrete citations enforce a hard cut-off to similarity. This is counter-intuitive to similarity-based learning, and ignores that scientific papers can be very similar despite lacking a direct citation – a core problem of finding related research. Instead, we use controlled nearest neighbor sampling over citation graph embeddings for contrastive learning. This control allows us to learn continuous similarity, to sample hard-to-learn negatives *and positives*, and also to avoid collisions between negative and positive samples by controlling the sampling margin between them. The resulting method SciNCL outperforms the state-of-the-art on the SciDocs benchmark. Furthermore, we demonstrate that it can train (or tune) models sample-efficiently, which improves compute efficiency, and that it can be combined with recent training-efficient methods. Perhaps surprisingly, even training a general-domain language model this way outperforms baselines pretrained in-domain.

1 Introduction

Pretrained language models (PLMs) achieve state-of-the-art results through fine-tuning on many NLP tasks (Rogers et al., 2020). However, the sentence or document embeddings derived from PLMs are of lesser quality compared to simple baselines like GloVe (Reimers and Gurevych, 2019), as their embedding space suffers from being anisotropic, i.e. poorly defined in some areas (Li et al., 2020).

One approach that has recently gained attention is the combination of PLMs with contrastive fine-tuning to improve the semantic textual similarity between document representations (Wu et al., 2020; Gao et al., 2021). These contrastive methods learn to distinguish between pairs of similar and dissimilar texts. As part of metric learning, they traditionally focused on defining new loss functions, while Musgrave et al. (2020) showed that newer metric losses lead to insignificant performance gains when compared fairly. Instead, recent works on self and supervised contrastive learning has started to focus on developing techniques that generate better positive and negative data augmentations for efficient contrastive learning (Tian et al., 2020; Rethmeier and Augenstein, 2021; Shorten et al., 2021).

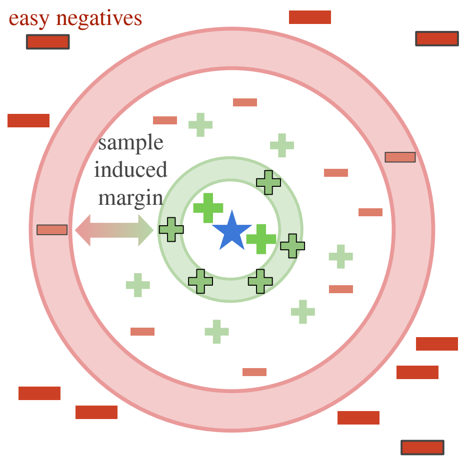


Figure 1: Starting from a query paper in a citation graph embedding space. Hard positives are citation graph embeddings that are sampled from a similar (close) context of, but are not so close that their gradients collapse easily. Hard (to classify) negatives (red band) are close to positives (green band) up to a *sampling induced margin*. Easy negatives are very dissimilar (distant) from the query paper.

In this paper, we focus on learning scientific document representations (SDRs). The core distinguishing feature of this domain is the presence of citation information. SDR methods like SciBERT (Beltagy et al., 2019) pretrain a Transformer on domain-specific text. The current state-of-the-art by Cohan et al. (2020) uses discrete citation information to generate positive and negative samples for contrastive fine-tuning of SciBERT via a triplet loss (Schroff et al., 2015). Cited papers are used to generate positive samples, while non-cited papers are negative samples.

This discrete cut-off to similarity is counter-intuitive to (continuous) similarity-based learning. It encourages overfitting to human similarity annotations, i.e. citations, which may reflect politeness and policy rather than semantic similarity (Pasternack, 1969). Such sample generation may also cause positive and negative samples to collide between cited papers, which Wang and Isola (2020) have shown to degrade contrastive optimization. Instead, the generation of *non-colliding* contrastive samples should be based on a continuous similarity function that allow us to find semantically similar papers, despite a lack of direct citation.

Contributions:

- We propose neighborhood contrastive learning for scientific document representations with citation graph embeddings (SciNCL).
- We sample similar and dissimilar papers from neighboring citation graph embeddings, such that both are hard to learn to avoid long training times and gradient collapse.
- As in recent contrastive learning works, we address sample generation semantics based on contrastive learning theory insights rather than designing new loss functions.
- We compare against the state-of-the-art approach SPECTER (Cohan et al., 2020) and other strong methods on the SciDocs benchmark and find that SciNCL outperforms SPECTER on average and on 9 of 12 tasks.
- Finally, we demonstrate that with SciNCL, using only 1% of the training data, starting with a general-domain language model, or training only the bias terms of the model is sufficient to outperform the baselines.
- Our code and models are publicly available.[1]

2 Related Work

Contrastive Learning pulls representations of similar data points (positives) closer together, while representations of dissimilar documents (negatives) are pushed apart. A common contrastive objective is the triplet loss (Schroff et al., 2015) that Cohan et al. (2020) used for scientific document representation learning, as we describe below. However, as Musgrave et al. (2020); Rethmeier and Augenstein (2021) point out, contrastive objectives work best when specific requirements are respected. **(Req. 1)** Views of the same data should introduce new information, i.e. the mutual information between views should be minimized (Tian et al., 2020). We use citation graph embeddings to generate contrast label information that supplements text-based similarity. **(Req. 2)** For training time and sample efficiency, negative samples should be hard to classify, but should also not collide with positives (Saunshi et al., 2019). **(Req. 3)** Recent works like Musgrave et al. (2020); Khosla et al. (2020) use multiple positives. However, positives need to be consistently close to each other (Wang and Isola, 2020), since positives and negatives may otherwise collide, e.g., Cohan et al. (2020) consider only 'citations by the query' as similarity signal and not 'citations to the query'. Such unidirectional similarity does not guarantee that a negative paper (not cited by the query) may cite the query paper and thus could cause collisions, the more we sample. Our method treats both citing and being cited as positives (Req. 2), while it also generates hard negatives and hard positives (Req. 2+3). Hard negatives are close to but do not overlap positives (red band in Fig. 1). Hard positives are close, but not trivially close to the query document (green band in Fig. 1).

Scientific Document Representations based on Transformers (Vaswani et al., 2017) and pretrained on domain-specific text dominate today's scientific document processing. There are SciBERT (Beltagy et al., 2019), BioBERT (Lee et al., 2019), or SciGPT2 (Luu et al., 2021), to name a few. Recent works modify these domain PLMs to support cite-worthiness detection (Wright and Augenstein, 2021), similarity (Ostendorff et al., 2020) or fact checking (Wadden et al., 2020).

Aside from text, citations are a common signal for the similarity of research papers. Paper (node) representations can be learned using the citation graph (Wu et al., 2019; Perozzi et al., 2014; Grover and Leskovec, 2016). Especially for recommendations of papers or citations, hybrid combinations of text and citation features are often employed (Han et al., 2018; Jeong et al., 2020; Molloy et al., 2020; Färber and Sampath, 2020).

Closest to SciNCL are Citeomatic (Bhagavatula et al., 2018) and SPECTER (Cohan et al., 2020).

[1] Anonymous `https://github.com/f4g2/s`

While Citeomatic relies on bag-of-words for its textual features, SPECTER is based on SciBERT. Both leverage citations to learn a triplet-based document embedding model, whereby positive samples are papers cited in the query. Easy negatives are random papers not cited by the query. Hard negatives are citations of citations – papers referenced in positive citations of the query, but are not cited directly by it. Citeomatic also uses a second type of hard negatives, which are the nearest neighbors of query a that are not cited by it.

Unlike our approach, Citeomatic does not use the neighborhood of citation embeddings, but instead relies on the actual document embeddings from the previous epoch. Despite being related to SciNCL, the sampling approaches employed in Citeomatic and SPECTER do not account for the pitfalls of using discrete citations as signal for paper similarity. Our work addresses this issue.

3 Methodology

Our goal is to learn task-independent representations for scientific documents. To do so we sample three document representation vectors and learn their similarity. For a given query paper vector $\boldsymbol{d}^Q$, we sample a positive (similar) paper vector $\boldsymbol{d}^+$ and a negative (dissimilar) paper vector $\boldsymbol{d}^-$. This produces a 'query, positive, negative' triple $(\boldsymbol{d}^Q, \boldsymbol{d}^+, \boldsymbol{d}^-)$ – represented by (★, ✚, ▬) in Fig. 1. To learn paper similarity, we need to define three components: (§3.1) how to calculate document vectors $\boldsymbol{d}$ for the loss over triplets $\mathcal{L}$; (§3.2) how citations provide similarity between papers; and (§3.3) how negative and positive papers $(\boldsymbol{d}^-, \boldsymbol{d}^+)$ are sampled as (dis-)similar documents from the neighborhood of a query paper $\boldsymbol{d}^Q$.

3.1 Contrastive Learning Objective

Given the textual content of a document d (paper), the goal is to derive a dense vector representation $\boldsymbol{d}$ that best encodes the document information and can be used in downstream tasks. A Transformer language model f (SciBERT; Beltagy et al. (2019)) encodes documents d into vector representations $f(d) = \boldsymbol{d}$. The input to the language model is the title and abstract separated by the `[SEP]` token.[2] The final layer hidden state of the `[CLS]` token is then used as a document representation $f(d) = \boldsymbol{d}$.

[2]Cohan et al. (2019) evaluated other inputs (venue or author) but found the title and abstract to perform best.

Training with a masked language modeling objectives alone has been shown to produce sub-optimal document representations (Li et al., 2020; Gao et al., 2021). Thus, similar to the SDR state-of-the-art method SPECTER (Cohan et al., 2020), we continue training the SciBERT model (Beltagy et al., 2019) using a self-supervised triplet margin loss (Schroff et al., 2015):

$$\mathcal{L} = \max\left\{ \|\boldsymbol{d}^Q - \boldsymbol{d}^+\|_2 - \|\boldsymbol{d}^Q - \boldsymbol{d}^-\|_2 + \xi, 0 \right\}$$

Here, ξ is a slack term ($\xi = 1$ as in SPECTER) and $\|\Delta \boldsymbol{d}\|_2$ is the L^2 norm, used as a distance function. However, the SPECTER sampling method has significant drawbacks. We will describe these issues and our contrastive learning theory guided improvements in detail below in §3.2.

3.2 Citation Neighborhood Sampling

Compared to the textual content of a paper, citations provide an outside view on a paper and its relation to the scientific literature (Elkiss et al., 2008), which is why citations are traditionally used as a similarity measure in library science (Kessler, 1963; Small, 1973). However, using citations as a discrete similarity signal, as done in Cohan et al. (2020), has its pitfalls. Their method defines papers cited by the query as positives, while paper citing the query could be treated as negatives. This means that *positive and negative learning information collides* between citation directions, which Wang and Isola (2020) have shown to deteriorate performance. Furthermore, a cited paper can have a low similarity with the citing paper given the many motivations a citation can have (Teufel et al., 2006). Likewise, a similar paper might not be cited.

To overcome these limitations, we learn citation embeddings first and then use the citation neighborhood around a given query paper d^Q to construct similar (positive) and dissimilar (negative) samples for contrast by using neighborhood information from either kNN (I) or a distance metric Sim (II-IV) as detailed in §3.3. This builds on the intuition that nodes connected by edges should be close to each other in the embedding space (Perozzi et al., 2014; Grover and Leskovec, 2016). Using citation embeddings allows us to: (1) sample paper similarity on a continuous scale, which makes it possible to: (2) define hard to learn positives, as well as (3) hard or easy to learn negatives. Points (2-3) are important in making contrastive learning efficiently as will describe below in §3.3.

3.3 Positives and Negatives Sampling

Positive samples d^+ should be semantically similar to the query paper d^Q, i.e. sampled close to the query embedding $\boldsymbol{d}^Q$. Additionally, as Wang and Isola (2020) find, positives should be sampled from comparable locations (distances from the query) in embedding space and be dissimilar enough from the query embedding, such that gradients do not collapse (become 0). Therefore, we sample positive (similar) papers within a narrow range $(k^+ - c^+, k^+]$ around the query vector, i.e. the green band in Fig. 1. When sampling from kNN neighbors, we use a small k^+ to find positives and later analyze the impact of k^+ in Fig. 2.

Negative samples can be divided into easy and hard negative samples (light and dark red in Fig. 1). Sampling more hard negatives is known to improve contrastive learning (Bucher et al., 2016; Wu et al., 2017). However, we make sure to sample hard negatives (red band in Fig. 1) such that they are close to potential positives but do not collide with positives (green band), by not sampling between them to 'induce a margin'. We do so, since Saunshi et al. (2019) showed that sampling a larger number of hard negatives only improves performance *if the negatives do not collide with positive samples*, since collisions make the learning signal noisy. That is, in the margin between hard negatives and positives we expect positives and negatives to collide, thus we avoid sampling from this region. To generate a broad self-supervised citation similarity signal for contrastive SDR learning, we also sample easy negatives that are farther from the query than hard negatives. For negatives, the k^- should be large when sampling via kNN, while the similarity threshold t^- should be small, to ensure samples are dissimilar from the query paper.

3.4 Sampling Strategies

As described in §3.2 and §3.3, our approach improves upon the method by Cohan et al. (2020). Therefore, we reuse their sampling parameters (5 triplets per query paper) and then further optimizing our methods' hyperparameters. For example, to train the triplet loss, we generate the same amount of $(\boldsymbol{d}^Q, \boldsymbol{d}^+, \boldsymbol{d}^-)$ triples per query paper as SPECTER (Cohan et al., 2020). To be precise, this means we generate $c^+{=}5$ positives (as explained in §3.3). We also generate 5 negatives, three easy negatives $c^-_{\text{easy}}{=}3$ and two hard negatives $c^-_{\text{hard}}{=}2$, as described in §3.3.

Below, we describe four strategies (I-IV) for sampling triplets. These either sample neighboring papers from citation embeddings (I-II), by random sampling (III), or using both strategies (IV). For each strategy, let c' be the number of samples for either positives c^+, easy negatives c^-_{easy}, or hard negatives c^-_{hard}.

Citation Graph Embeddings: We train a graph embedding model f_c on citations extracted from the Semantic Scholar Open Research Corpus (S2ORC; Lo et al. (2020)) to get citation embeddings C. S2ORC contains 52.6M nodes (papers) and 467K edges (citations). At this scale, many existing graph embedding frameworks require substantial computing resources. Hence, we utilize PyTorch BigGraph (Lerer et al., 2019), which allows for training with modest hardware requirements. Our method performs well using the default training settings from Lerer et al. (2019), but given more computational resources, careful tuning may produce even better-performing graph embeddings. Nonetheless, we conducted a narrow parameter search using the S2ORC link prediction task – see Appendix A.2.

(I) K-nearest neighbors (kNN): Assuming a given citation embedding model f_c and a search index (e.g., FAISS §4.3), we run $KNN(f_c(d^Q), C)$ and take c' samples from a range of the $(k - c', k]$ nearest neighbors around the query paper d^Q with its neighbors $N{=}\{n_1, n_2, n_3, \dots\}$, whereby neighbor n_i is the i-th nearest neighbor citation. For instance, for $c'{=}3$ and $k{=}10$ the corresponding samples would be the three neighbors descending from the tenth neighbor: n_8, n_9, and n_{10}. In practice, we sample the neighbors N only once via $[0; \max(k)]$, and then generate triples by range-selection in N; i.e. positives $= (k^+ - c^+; k^+]$, and hard negatives $= (k^-_{\text{hard}} - c^-_{\text{hard}}; k^-_{\text{hard}}]$.

(II) Similarity threshold (Sim): Take c' papers that are within the similarity threshold t of a query paper d^Q such that $s(f_c(d^Q), f_c(d_i)) < t$, where s is the cosine similarity function. For example, given the similarity scores $S{=}\{0.9, 0.8, 0.7, 0.1\}$ (ascending order, the higher the similarity is the closer the candidate embedding to the query embedding is) with $c'{=}2$ and $t{=}0.5$, the two candidates with the largest similarity scores and smaller than the threshold would be 0.8 and 0.7. The corresponding papers would be selected as samples.

(III) Random sampling Sample any c' papers without replacement from the corpus.

(IV) Filtered random Like (III) but excluding the papers that are retrieved by kNN or SIM, i.e., all neighbors within the largest k or n_i with $i{<=}k$ are excluded.

The kNN and SIM sampling strategies introduce hyperparameters (k or t) that allow for the *controlled sampling of positives or negatives* with different difficulty (from easy to hard depending on the hyperparameter). Specifically, in Fig. 1 these hyperparameters define the tunable *sample induced margin* between positives and negatives, as well as the width and position of the positive sample band (green) and negative sample band (red) around the query sample. Besides the strategies above, we experiment with k-means clustering and sorted random sampling, neither of which performs well (see negatives results in Appendix A.3).

4 Experiments

We next introduce our experimental setting including the data used for training and evaluation, as well as implementation details.

4.1 Evaluation Dataset

We evaluate the document representations on the SCIDOCS benchmark (Cohan et al., 2020). A key difference to other benchmarks is that embeddings are the input to the individual tasks without explicit fine-tuning. The SCIDOCS benchmark consists of the following four tasks:

Document classification (CLS) with labels from Medical Subject Headings (MeSH) (Lipscomb, 2000) and Microsoft Academic Graph (MAG) (Sinha et al., 2015) evaluated with the F1 metric. **Co-views and co-reads** (USR) prediction based on the L2 distance between embeddings. Co-views are papers viewed in a single browsing session. Co-read refers to a user accessing the PDF of a paper. Both user activities are evaluated using Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG). **Direct and co-citation** (CITE) prediction based on the L2 distance between the embeddings. MAP and nDCG are the evaluation metrics. **Recommendations** (REC) generation based on embeddings and paper metadata to rank a set of "similar papers" for a given paper. An offline evaluation with historical clickthrough data determines the performance using Precision@1 (P@1) and nDCG.

4.2 Training Data

We replicate the training data from SPECTER as closely as possible. Unfortunately SPECTER's data is only provided as triples of Semantic Scholar paper IDs (Ammar et al., 2018). To obtain paper title, abstract, and citations, we try mapping SPECTER's papers to S2ORC. We successfully map 96.1% of the query papers and 69.3% of the corpus from which positives and negatives are sampled. To account for the missing papers, we randomly sample papers from S2ORC such that the absolute number of papers is identical with SPECTER. The SCIDOCS papers are excluded. The ratio of training triples per query remains the same (§3.4).

4.3 Training and Implementation

We replicate the training setup from SPECTER as closely as possible. We implement SciNCL using Huggingface Transformers (Wolf et al., 2020), initialize the model with SciBERT's weights (Beltagy et al., 2019), and train via the triplet loss (Equation 3.1). The optimizer is Adam with weight decay (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) and learning rate $\lambda{=}2^{-5}$. To explore the effect of compute efficient fine-tuning we also train a BitFit model (Zaken et al., 2021) with $\lambda{=}1^{-4}$ (§7.2). We train SciNCL on two NVIDIA GeForce RTX 6000 (24G) for 2 epochs (approx. 24 hours of training time) with batch size 8 and gradient accumulation for an effective batch size of 32 (same as SPECTER). Training the S2ORC graph embeddings takes approx. 6 hours. The kNN and SIM strategies are implemented with FAISS (Johnson et al., 2021) using a flat index (exhaustive search) and take less than 30min to compute.

4.4 Baseline Methods

We compare to 10 prior approaches: Doc2Vec (Le and Mikolov, 2014), weighted sum of in-domain fastText word embeddings (Bojanowski et al., 2017), averaged contextualized token-level representations from ELMO (Peters et al., 2018), BERT (Devlin et al., 2019) a state-of-the-art PLM pretrained on general-domain text, BioBERT-Base-Cased-v1.2 (Lee et al., 2019) a BERT variations for biomedical text, SciBERT (Beltagy et al., 2019) a BERT variation for scientific text, CiteBERT (Wright and Augenstein, 2021) a SciBERT variation fine-tuned on cite-worthiness detection,

Task →	Classification		User activity prediction				Citation prediction				Recomm.		Avg.
Subtask →	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model ↓ / Metric →	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
Doc2Vec* (2014)	66.2	69.2	67.8	82.9	64.9	81.6	65.3	82.2	67.1	83.4	51.7	16.9	66.6
fastText-sum* (2017)	78.1	84.1	76.5	87.9	75.3	87.4	74.6	88.1	77.8	89.6	52.5	18.0	74.1
ELMo* (2018)	77.0	75.7	70.3	84.3	67.4	82.6	65.8	82.6	68.5	83.8	52.5	18.2	69.0
Citeomatic* (2018)	67.1	75.7	81.1	90.2	80.5	90.2	86.3	94.1	84.4	92.8	52.5	17.3	76.0
SGC* (2019)	76.8	82.7	77.2	88.0	75.7	87.5	91.6	96.2	84.1	92.5	52.7	18.2	76.9
BERT (2019)	79.9	74.3	59.9	78.3	57.1	76.4	54.3	75.1	57.9	77.3	52.1	18.1	63.4
SciBERT* (2019)	79.7	80.7	50.7	73.1	47.7	71.1	48.3	71.7	49.7	72.6	52.1	17.9	59.6
BioBERT (2019)	77.2	73.0	53.3	74.0	50.6	72.2	45.5	69.0	49.4	71.8	52.0	17.9	58.8
CiteBERT (2021)	78.8	74.8	53.2	73.6	49.9	71.3	45.0	67.9	50.3	72.1	51.6	17.0	58.8
SPECTER* (2020)	**82.0**	86.4	83.6	91.5	84.5	92.4	88.3	94.9	88.1	94.8	**53.9**	**20.0**	80.0
SciNCL (ours)	81.5	**88.8**	**85.5**	**92.4**	**87.6**	**93.9**	**93.2**	**97.1**	**91.6**	**96.4**	53.6	19.3	**81.8**
± σ w/ ten seeds	*.497*	*.125*	*.166*	*.101*	*.247*	*.153*	*.597*	*.26*	*.325*	*.147*	*.337*	*.626*	*.172*

Table 1: Results on the SciDocs benchmark. Our approach surpasses the previous best avg. score by 1.8 points and also outperforms the baselines in 9 of 12 task metrics. Our scores are reported as mean and standard deviation σ over ten random seeds. Baseline scores with * are taken from Cohan et al. (2020).

the graph-convolution approach SGC (Wu et al., 2019), Citeomatic (Bhagavatula et al., 2018), and SPECTER (Cohan et al., 2020). If not otherwise mentioned, all BERT variations are used in their `base-uncased` versions.

5 Overall Results

Tab. 1 shows our main results, comparing SciNCL with the best validation performance against prior approaches. SciNCL achieves an average performance of 81.8 across all metrics, which is a 1.8 point absolute improvement over the next-best baseline. We find the best validation performance when positives and hard negative are sampled with kNN, whereby positives are k^{+}=25, and hard negatives are k^{-}_{hard}=4000 (§6). Easy negatives are generated through filtered random sampling. As random sampling accounts for a large fraction of the triples (in the form of easy negatives), we report the mean scores and standard deviation based on ten random seeds (seed $\in [0, 9]$).

For MAG classification, SPECTER achieves the best result with 82.0 F1 followed by SciNCL with 81.5 F1 (-0.5 points). For MeSH classification, SciNCL yields the highest score with 88.8 F1 (+2.4 compared to SPECTER). Both classification tasks have in common that the chosen training settings lead to over-fitting. Changing the training by using only 10% training data, SciNCL yields 82.4 F1@MAG (Tab. 2). In all user activity and citation tasks, SciNCL yields higher scores than all baselines. It is notable that SciNCL also outperforms SGC on direct citation prediction, where SGC outperforms SPECTER in terms of nDCG.

On the recommender task, SPECTER yields the best nDCG and P@1, whereas SciNCL is slightly worst with 53.6 nDCG and 19.3 P@1 (-0.3 nDCG and -0.7 P@1 compared to SPECTER). The recommendation task shows the strongest effect of random seeds (σ of 0.3 nDCG and 0.6 P@1). The performance difference between SciNCL and SPECTER is close to or within the standard deviation. Hence, it remains unclear whether the difference is significant, since Cohan et al. (2019) do not report standard deviations. In contrast to the classification tasks, training for more than two epochs leads to further improvement on the recommendation task (currently under-fitting). As a result, one should adjust the training settings accordingly when aiming only for this particular task.

Regarding the PLM baselines, we observe that the general-domain BERT, with a score of 63.4, outperforms the domain-specific BERT variants, namely SciBERT (59.6), BioBERT (58.8), and CiteBERT (58.8). Still, all PLMs without contrastive objectives yield substantially worse results (even compared to Doc2Vec or fastText). This emphasizes the anisotropy problem of embeddings directly extracted from current PLMs.

In summary, we show that SciNCL's triple selection on average leads to an improved performance on SciDocs, with most gains being observed for user activity and citation tasks. Examples of the generated triples are shown in Appendix A.5.

6 Impact of Sample Difficulty

The benefit of SciNCL is that the hyperparameters of the sampling strategies can be tuned (§3.3) to learn without sample collisions. In this section, we present the results of this tuning procedure. We optimize the sampling strategies for positives and negatives with partial grid search on a random sample of 10% of the original training data (sampling based on queries). Our experiments show that optimizations on this subset correlate with the entire dataset. The scores in Fig. 2 and 3 are reported as the mean over three random seeds including standard deviations.

6.1 Positive Samples

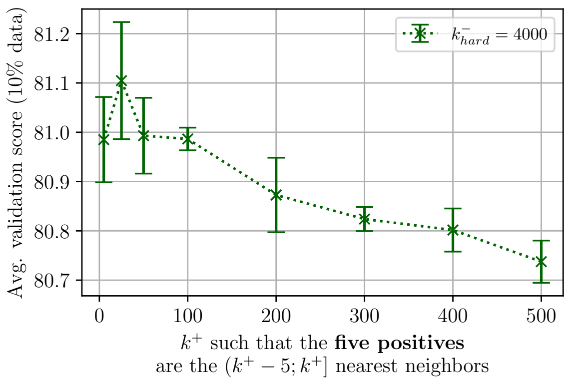


Figure 2: Results on the validation set w.r.t. positive sampling with kNN when using 10% training data.

Fig. 2 shows the average scores on the SciDocs validation set depending on the selection of positives with the kNN strategy. We only change k^+, while negative sampling remains fixed to its best setting (§6.2). The Sim strategy is omitted for positive sampling since it yields a poor performance throughout all tasks (Appendix A.3)

The performance is relatively stable for $k^+ < 100$ with peak at $k^+ = 25$, for $k^+ > 100$ the performance declines as k^+ increases. Wang and Isola (2020) state that positive samples should be semantically similar to each other, but not too similar to the query. For example, at $k^+ = 5$, positives may be a bit "too easy" to learn, such that they produce less informative gradients than the optimal setting $k^+ = 25$. Similarly, making k^+ too large leads to the *sampling induced margin* being too small, such that positives collide with negative samples, which creates contrastive label noise that degrades performance Saunshi et al. (2019).

Another observation is the standard deviation σ: One would expect σ to be independent of k^+ since random seeds affect only the negatives. However, positives and negatives interact with each other through the triplet margin loss. Therefore, σ is also affected by k^+.

6.2 Hard Negative Samples

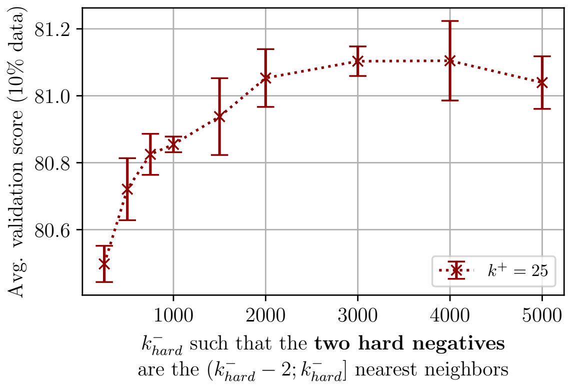


Figure 3: Results on the validation set w.r.t. hard negative sampling with kNN using 10% training data.

Fig. 3 presents the validation results with kNN strategy and k^-_{hard} and the best setting for positives ($k^+ = 25$). The performance increases with increasing k^-_{hard}, until the performance plateaus for $2000 < k^-_{\text{hard}} < 4000$ with a peak at $k^-_{\text{hard}} = 4000$. This plateau can also be observed in the test performance, where $k^-_{\text{hard}} = 2000$ and $k^-_{\text{hard}} = 3000$ yield a marginally lower score of 81.7 (Tab. 2). For $k^-_{\text{hard}} > 4000$, the performance starts to decline again. This suggests that for large k^-_{hard} the samples are not "hard enough". The need for hard negatives confirms the findings of Cohan et al. (2020).

Intuitively, the kNN strategy should suffer from a centrality or hubness problem. How many neighbors are semantically similar strongly depends on the query paper itself. A popular and frequently cited paper has many more similar neighbors than a niche paper. To test this assumption, we also evaluate the Sim strategy that should account for the hubness problem. However, Sim underperforms with a score of 81.5 (Tab. 2) independent from different similarity thresholds (Appendix A.3).

6.3 Easy Negative Samples

Filtered random sampling of easy negatives yields the best validation performance compared pure random sampling (Tab. 2). However, the performance

	CLS	USR	CITE	REC	Avg.	Δ
SciNCL	85.0	**89.0**	**94.7**	36.5	**81.8**	–
SPECTER	84.2	88.4	91.5	36.9	80.0	-1.8
k^{-}_{hard}=2000	85.1	**88.9**	**94.7**	36.3	81.7	-0.1
k^{-}_{hard}=3000	84.7	88.8	**94.7**	36.2	81.7	-0.1
hard neg. w/ SIM	84.4	88.8	94.5	35.8	81.5	-0.2
easy neg. w/ random	85.2	**88.9**	**94.7**	36.5	**81.8**	0.0
Init. w/ BERT-Base	84.2	88.5	93.9	37.3	81.3	-0.5
Init. w/ BERT-Large	85.0	88.7	94.1	36.3	81.5	-0.3
Init. w/ BioBERT	84.2	88.8	93.9	**37.8**	81.5	-0.3
1% training data	85.6	88.2	92.6	36.1	80.8	-1.0
10% training data	**85.9**	88.7	93.7	36.3	81.4	-0.8
BitFit training	85.8	88.7	93.7	35.7	81.3	-0.5

Table 2: Ablations. Numbers are averages of metrics for each task of the SciDocs test set, average score over all metrics, and absolute difference to SciNCL.

difference is marginal. When rounded to one decimal, their average test scores are identical. The marginal difference is caused by the large corpus size and the resulting small probability of randomly sampling one paper from the KNN results. But without filtering, the effect of random seeds increases, since we find a higher standard deviation compared to the one with filtering.

As a potential way to decrease randomness, we experiment with other approaches like k-means clustering but find that they decrease the performance (Appendix A.3).

7 Ablation Analysis

In addition to sample difficulty, we evaluate the performance impact of data quantity, trainable parameters, and language model initialization.

7.1 Initial Language Models

Tab. 2 shows the effect of initializing the model weights not with SciBERT but with general-domain PLMs (BERT-Base and BERT-Large) or with BioBERT. The initialization with other PLMs decreases the performance. However, the decline is marginal (BERT-Base -0.5, BERT-Large -0.3, BioBERT -0.3) and all PLMs outperform the SPECTER baseline. For the recommendation task, in which SPECTER is superior over SciNCL, BioBERT and BERT-Base both outperform SPECTER. This indicates that the improved triple mining of SciNCL has a greater domain adaption effect than pretraining on domain-specific literature. Given that pretraining of PLMs requires a magnitude more resources than the fine-tuning with SciNCL, our approach can be a solution for resource-limited use cases.

7.2 Data and Compute Efficiency

The last three rows of Tab. 2 show the results regarding data and compute efficiency. Training SciNCL with only 10% of the original data yields a score of 81.4 (-0.8 points). Even with only 1% training data (7300 triples), SciNCL achieves a score of 80.8 that is 1.0 points less than with 100% but still 0.8 points more than the SPECTER baseline. With this data efficiency, one could manually create a triplet dataset or use existing expert-annotated datasets like Brown et al. (2019).

Lastly, we evaluate BitFit training (Zaken et al., 2021), which only trains the bias terms of the model while freezing all other parameters. This corresponds to training only 0.1% of the original parameters. With BitFit, SciNCL yields a considerable score of 81.3 (-0.5 points). As a result, SciNCL could be trained on the same hardware with even larger (general-domain) language models (§7.1).

8 Conclusion

We present a novel approach for contrastive learning of scientific document embeddings that addresses the challenge of selecting informative positive and negative samples. By leveraging citation graph embeddings for sample generation, SciNCL achieves a score of 81.8 on the SciDocs benchmark, a 1.8 point improvement over the previous best method SPECTER. This is purely achieved by introducing tunable sample difficulty and avoiding collisions between positive and negative samples, while existing PLM and data setups can be reused.

Our work highlights the importance of sample generation in a contrastive learning setting. We show that 1% of training data is already sufficient to outperform SPECTER, whereas the remaining 99% provide only 1.0 additional points (80.8 to 81.8). We also demonstrate that in-domain language model pretraining (like SciBERT) is beneficial, while general-domain PLMs can achieve a comparable performance and even outperform SPECTER. This indicates that controlling sample difficulty and avoiding collisions is more effective than in-domain pretraining, especially in scenarios where training a PLM from scratch is infeasible.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine Van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 3:84–91.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. 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)*, pages 3613–3618, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:238–251.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Peter Brown, Y. Zhou, and Darius Widera. 2019. Large expert-curated database for benchmarking document similarity detection in biomedical literature search. *Database*, pages 1–66.

Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. 2016. Hard negative mining for metric learning based zero-shot classification. In *Computer Vision – ECCV 2016 Workshops*, pages 524–531, Cham. Springer International Publishing.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *Proceedings of the 2019 Conference of the North*, volume 1, pages 3586–3596, Stroudsburg, PA, USA. Association for Computational Linguistics.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62.

Michael Färber and Ashwath Sampath. 2020. HybridCite: A Hybrid Model for Context-Aware Citation Recommendation. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 117–126, New York, NY, USA. ACM.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv:2104.08821*.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 855–864, New York, New York, USA. ACM Press.

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. 2018. hyperdoc2vec: Distributed Representations of Hypertext Documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2384–2394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chanwoo Jeong, Sion Jang, Hyuna Shin, Eunjeong Lucy Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, pages 1–16.

Jeff Johnson, Matthijs Douze, and Herve Jegou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

M. M. Kessler. 1963. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on Machine Learning*, 32:1188–1196.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, pages 1–8.

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of The Conference on Systems and Machine Learning*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019.*

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. Explaining Relationships Between Scientific Documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144, Stroudsburg, PA, USA. Association for Computational Linguistics.

Paul Molloy, Joeran Beel, and Akiko Aizawa. 2020. Virtual Citation Proximity (VCP): Learning a Hypothetical In-Text Citation-Proximity Metric For Uncited Documents. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*, pages 1–8.

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXV*, pages 681–699.

Malte Ostendorff, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. 2020. Aspect-based Document Similarity for Research Papers. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020).*

Simon Pasternack. 1969. The scientific enterprise: Public knowledge. an essay concerning the social dimension of science. *Science*, 164(3880):669–670.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 701–710, New York, New York, USA. ACM Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nils Rethmeier and Isabelle Augenstein. 2021. A primer on contrastive pretraining in language processing: Methods, lessons learned and perspectives. *arXiv:2102.12982*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, volume 97 of *PMLR*. PMLR.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823.

Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *J. Big Data*, 8(1):101.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. *Proceedings of the 24th International Conference on World Wide Web*.

Henry Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*, page 103, Morristown, NJ, USA. Association for Computational Linguistics.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Dustin Wright and Isabelle Augenstein. 2021. Cite-Worth: Cite-Worthiness Detection for Improved Scientific Document Understanding. In *Findings of ACL-IJCNLP*. Association for Computational Linguistics.

Chao-yuan Wu, R. Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling Matters in Deep Embedding Learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2859–2867. IEEE.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 6861-6871, pages 815–826. PMLR.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive Learning for Sentence Representation. *arXiv:2012.15466*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. *arXiv:2106.10199*.

A Appendix

A.1 Citation Data

The version identifier of S2ORC is `20200705v1`. The full citation graph consists of 52.6M nodes (papers) and 467K edges (citations).

A.2 Graph Embedding Evaluation

To evaluate the underlying citation graph embeddings, we experiment with a few of BigGraph's hyperparameters. We trained embeddings with different dimensions $d=\{128, 512, 768\}$ and different distance measures (cosine similarity and dot product) on 99% of the data and test the remaining 1% on the link prediction task. An evaluation of the graph embeddings with SciDocs is not possible since we could not map the papers used in SciDocs to the S2ORC corpus. All variations are trained for 20 epochs, margin $m=0.15$, and learning rate $\lambda=0.1$ (based on the recommended settings by Lerer et al. (2019)).

Table 3: Link prediction performance of BigGraph embeddings trained on S2ORC citation graph with different dimensions and distance measures.

Dim.	Dist.	MRR	Hits@1	Hits@10	AUC
128	Cos.	54.09	43.39	75.21	85.75
128	Dot	89.75	85.84	96.13	97.70
512	Dot	94.60	92.47	97.64	98.64
768	Dot	95.12	93.22	97.77	98.74

Tab. 3 shows the link prediction performance measured in MRR, Hits@1, Hits@10, and AUC. Dot product is substantially better than cosine similarity as distance measure. Also, there is a positive correlation between the performance and the size of the embeddings. The larger the embedding size the better link prediction performance. Graph embeddings with $d=768$ were the largest possible size given our compute resources (available disk space was the limiting factor).

A.3 Negative Results

We tried additional sampling strategies and model modification of which none led to an performance improvement.

kNN with interval large than c Our best results are achieved with kNN where the size of the neighbor interval $(k - c'; k]$ is equal to the number of samples c' that the strategy should generate. In addition to this, we also experimented with large intervals, e.g., $(1000; 2000]$, from which c' papers are randomly sampled. This approach yields comparable results but suffers from a larger effect of randomness and is therefore more difficult to optimize.

K-Means Cluster for Easy Negatives Easy negatives are supposed to be far away from the query. Random sampling from a large corpus ensures this as our results show. As an alternative approach, we tried k-means clustering whereby we selected easy negatives from the centroid that has a given distance to the query's centroid.

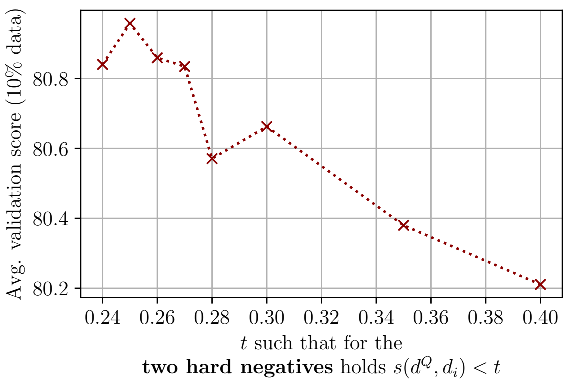


Figure 4: Results on the validation set w.r.t. hard negative sampling with Sim using 10% training data.

Hard Negatives With Similarity Threshold As shown in Tab. 2, hard negative sampling with k nearest neighbors outperforms absolute similarity sampling. Fig. 4 show the validation results for different similarity thresholds. A similar pattern as in Fig. 3 can be seen. When the negatives are closer to the query paper (larger similarity threshold t), the validation score decreases.

Positives with Similarity Threshold Positive sampling with Sim performs poorly since even for small $t^+ < 0.5$ many query papers do not have any neighbors within this similarity threshold (more than 40%). Solving this issue would require changing the set of query papers which we omit for comparability to SPECTER.

Sorted Random Simple random sampling does not ensure if a sample is far or close to the query. To integrate a distance measure in the random sampling, we first sample n candidates, then order the candidates according to their distance to the query, and lastly select the c' candidates that are the closest or furthest to the query as samples.

Mask Language Modeling Giorgi et al. (2021) show that combining a contrastive loss with a mask language modeling loss can improve text representation learning. However, in our experiments a combined function decreases the performance on SciDocs, probably due to the effects found by (Li et al., 2020).

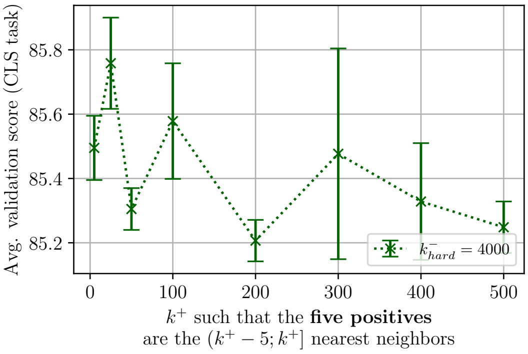
Graph Embedding Prediction Loss We combine the triplet loss (Equation 3.1) with a MSE loss of the predicted embedding and the graph embeddings. This approach yields a comparable performance but adds additional computational complexity and was therefore discarded for the final experiments.

A.4 Task-specific Results

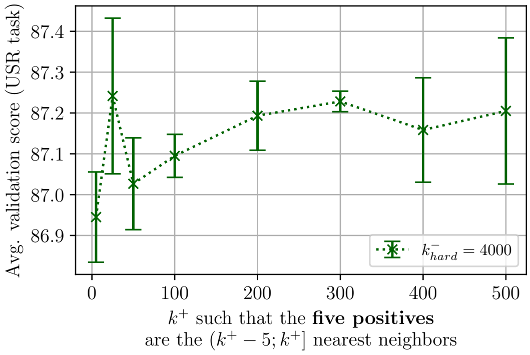
Fig. 5 and 6 present the validation performance like in §6 but on a task-level and not as an average over all tasks. The plots show that the optimal k^+ and k^-_{hard} values are partially task dependent.

A.5 Examples

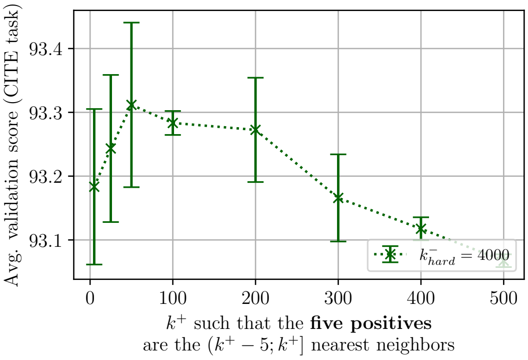
Tab. 4 lists three examples of query papers with their corresponding positive and negative samples. The complete set of triples that we use during training are available in our code repository[1].



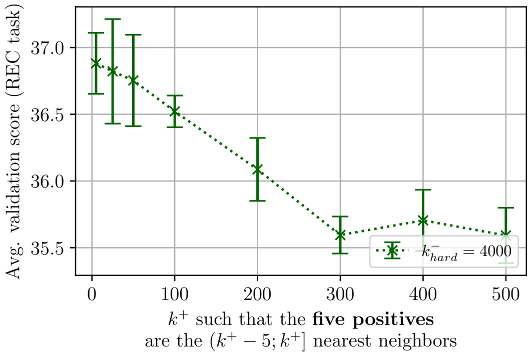
(a) Classification



(b) User activities

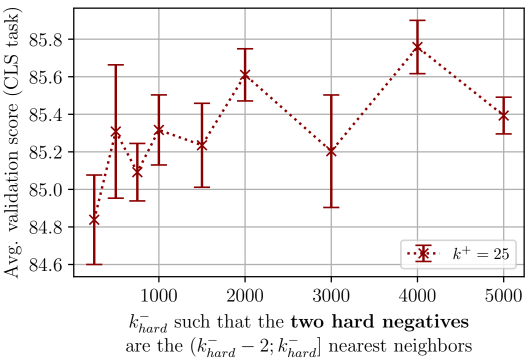


(c) Citation

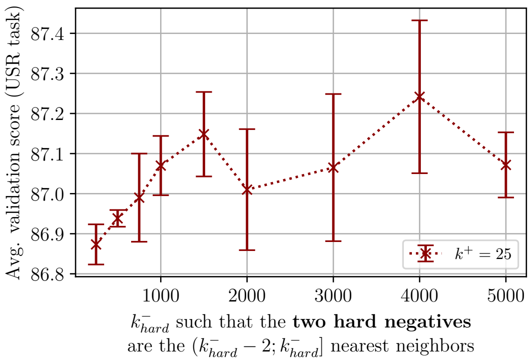


(d) Recommendation

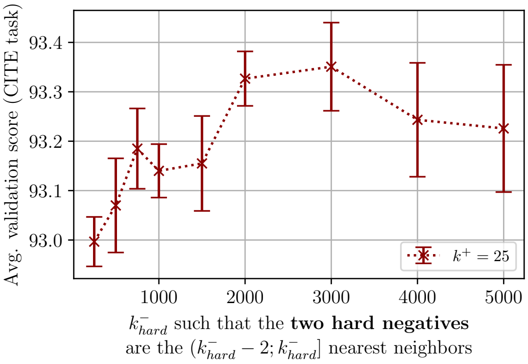
Figure 5: Task-level validation performance w.r.t. k^+ with KNN strategy using 10% training data.



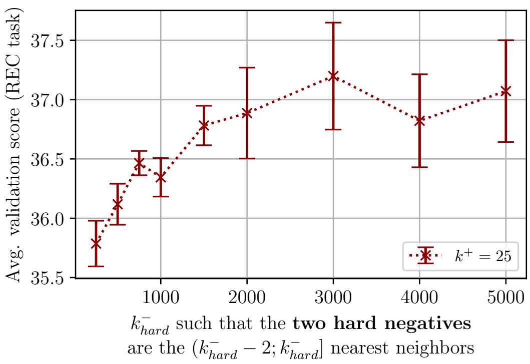
(a) Classification



(b) User activities



(c) Citation



(d) Recommendation

Figure 6: Task-level validation performance w.r.t. k^-_{hard} with KNN strategy using 10% training data.

Table 4: Example query papers with their positive and negative samples.

Query:	**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
Positives:	• A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference • Looking for ELMo's Friends: Sentence-Level Pretraining Beyond Language Modeling • GLUE : A MultiTask Benchmark and Analysis Platform for Natural Language Understanding • Dissecting Contextual Word Embeddings: Architecture and Representation • Universal Transformers
Negatives:	• Planning for decentralized control of multiple robots under uncertainty • Graph-Based Relational Data Visualization • Linked Stream Data Processing • Topic Modeling Using Distributed Word Embeddings • Adversarially-Trained Normalized Noisy-Feature Auto-Encoder for Text Generation
Query:	**BioBERT: a pre-trained biomedical language representation model for biomedical text mining**
Positives:	• Exploring Word Embedding for Drug Name Recognition • A neural joint model for entity and relation extraction from biomedical text • Event Detection with Hybrid Neural Architecture • Improving chemical disease relation extraction with rich features and weakly labeled data • GLUE : A MultiTask Benchmark and Analysis Platform for Natural Language Understanding
Negatives:	• Weakly Supervised Facial Attribute Manipulation via Deep Adversarial Network • Applying the Clique Percolation Method to analyzing cross-market branch banking ... • Perpetual environmentally powered sensor networks • Labelling strategies for hierarchical multi-label classification techniques • Domain Aware Neural Dialog System
Query:	**A Context-Aware Citation Recommendation Model with BERT and Graph Convolutional Networks**
Positives:	• Content-based citation analysis: The next generation of citation analysis • ScisummNet: A Large Annotated Dataset and Content-Impact Models for Scientific Paper ... • Citation Block Determination Using Textual Coherence • Discourse Segmentation Of Multi-Party Conversation • Argumentative Zoning for Improved Citation Indexing
Negatives:	• Adaptive Quantization for Hashing: An Information-Based Approach to Learning ... • Trap Design for Vibratory Bowl Feeders • Software system for the Mars 2020 mission sampling and caching testbeds • Applications of Rhetorical Structure Theory • Text summarization for Malayalam documents — An experience