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 cutoff 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) language models sample-efficiently 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

Large pretrained language models (LLMs) 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 LLMs 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 LLMs 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 (positive and negative samples). As recent works show (Tian et al., 2020b; Rethmeier and Augenstein, 2021a,b; Shorten et al., 2021), the selection of these positive and negative samples is crucial for efficient contrastive learning.

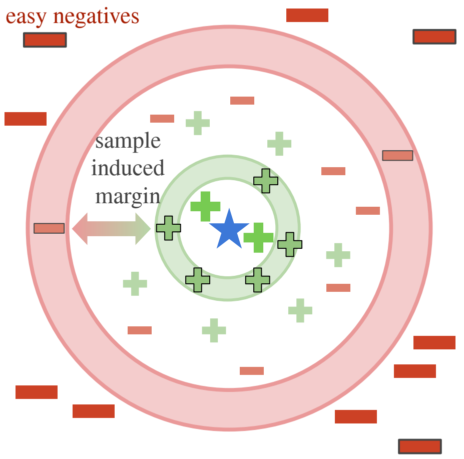


Figure 1: Starting from a query paper ★ in a citation graph embedding space. Hard positives ✚ are citation graph embeddings that are sampled from a similar (close) context of ★, but are not so close that their gradients collapse easily. Hard (to classify) negatives ▭ (red band) are close to positives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

This paper focusses on learning scientific document representations (SDRs). The core distinguishing feature of this domain is the presence of citation information that complement the textual information. The current state-of-the-art SPECTER by Cohan et al. (2020) uses citation information to generate positive and negative samples for contrastive fine-tuning of a SciBERT language model (Beltagy et al., 2019). SPECTER relies on 'citations by the query paper' as a discrete signal for similarity, i.e., positive samples are cited by the query while negative ones are not cited.

However, SPECTER's use of citations has its pitfalls. Considering only one citation direction may cause positive and negative samples to collide since a paper pair could be treated as a positive and negative instance simultaneously. Also, relying on a single citation as a discrete similarity signal is subject to noise, e.g., citations may reflect politeness and policy rather than semantic similarity (Pasternack, 1969) or related papers lack a direct citation (Gipp and Beel, 2009). This discrete cut-off to similarity is counter-intuitive to (continuous) similarity-based learning.

Instead, the generation of non-colliding contrastive samples should be based on a continuous similarity function that allows us to find semantically similar papers, even without direct citations. With SciNCL, we address these issues by generating contrastive samples based on citation embeddings. The citation embeddings, which incorporate the full citation graph, provide a continuous, undirected, and less noisy similarity signal that allows the generations of arbitrary difficult-to-learn positive and negative samples.

Contributions:

- We propose neighborhood contrastive learning for scientific document representations with citation graph embeddings (SciNCL) based on contrastive learning theory insights.
- We sample positive (similar) and negative (dissimilar) papers from the k nearest neighbors in the citation graph embedding space, such that positives and negatives do not collide but are also hard to learn.
- 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 metrics.
- Finally, we demonstrate that with SciNCL, using only 1% of the triplets for training, 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) 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., 2020b). 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 (Appendix A.7.10). 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). Appendix A.1 presents related work on triplet mining.

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) and SciGPT2 (Luu et al., 2021), to name a few. Recent works modify these domain LLMs to support cite-worthiness detection (Wright and Augenstein, 2021) or fact checking (Wadden et al., 2020).

Aside from text, citations are a valuable 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; Brochier et al., 2019; Yang et al., 2015; Holm et al., 2022).

[1] https://anonymous.4open.science/r/scincl-1553/

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

Cross-Modal Transfer. SciNCL transfers knowledge across modalities, i.e., from citations into a language model. According to Cohan et al. (2020), SciNCL can be considered as a "*citation-informed Transformer*". This cross-modal transfer learning is applied for various modalities (see Kaur et al. (2021) for an overview): text-to-image (Socher et al., 2013), RGB-to-depth image (Tian et al., 2020a), or graph-to-image (Wang et al., 2018). While the aforementioned methods incorporate cross-modal knowledge through joint loss functions or latent representations, SciNCL transfers knowledge through the contrastive sample selection, which we found superior to the direct transfer approach (Appendix A.7.9).

3 Methodology

Our goal is to learn citation-informed 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' triplet $(\boldsymbol{d}^Q, \boldsymbol{d}^+, \boldsymbol{d}^-)$ – 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 $f(d) = \boldsymbol{d}$.

Training with a masked language modeling objectives alone has been shown to produce suboptimal 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 Saunshi et al. (2019) 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.

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

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 the k nearest neighbors. This builds on the intuition that nodes connected by edges should be close to each other in the embedding space (Perozzi et al., 2014). 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 efficient 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, to avoid gradient collapse (zero gradients). Therefore, we sample c^+ positive (similar) papers from a close neighborhood around query embedding $\boldsymbol{d}^Q$ $(k^+ - c^+, k^+]$, i.e. the green band in Fig. 1. When sampling with kNN search, 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 using a tunable 'sampling induced 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 diverse 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 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 optimize our methods' hyperparameters. For example, to train the triplet loss, we generate the same amount of $(\boldsymbol{d}^Q, \boldsymbol{d}^+, \boldsymbol{d}^-)$ triplets 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 three strategies (I-III) for sampling triplets. These either sample neighboring papers from citation embeddings (I), by random sampling (II), or using both strategies (III). 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. We utilize PyTorch BigGraph (Lerer et al., 2019), which allows for training on large graphs with modest hardware requirements. The resulting graph embeddings perform well using the default training settings from Lerer et al. (2019), but given more computational resources, careful tuning may produce even better-performing embeddings. Nonetheless, we conducted a narrow parameter search based on link prediction – see Appendix A.5.

(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 in the citation embedding space. 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}. To reduce computing effort, we sample the neighbors N only once via $[0; \max(k^+, k^-_{\text{hard}})]$, and then generate triplets by range-selection in N; i.e. positives $= (k^+ - c^+; k^+]$, and hard negatives $= (k^-_{\text{hard}} - c^-_{\text{hard}}; k^-_{\text{hard}}]$.

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

(III) Filtered random: Like (II) but excluding the papers that are retrieved by kNN, i.e., all neighbors within the largest k are excluded.

The kNN sampling introduces the hyperparameter k that allows for the *controlled sampling of positives or negatives* with different difficulty (from easy to hard depending on k). Specifically, in Fig. 1 the hyperparameter k defines 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 similarity threshold, k-means clustering and sorted random sampling, neither of which performs well (Appendix A.7).

4 Experiments

4.1 Evaluation Dataset

We evaluate 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 Medical Subject Headings (MeSH) (Lipscomb, 2000) and Microsoft Academic Graph labels (MAG) (Sinha et al., 2015). **Co-views and co-reads** (USR) prediction based on the L2 distance between embeddings. **Direct and co-citation** (CITE) prediction based on the L2 distance between the embeddings. **Recommendations** (REC) generation based on embeddings and paper metadata.

4.2 Training Datasets

The experiments mainly compare SciNCL against SPECTER on the SciDocs benchmark. However, we found 40.5% of SciDocs's papers leaking into SPECTER's training data (the leakage affects only the unsupervised paper data but not the gold labels – see Appendix A.3). To be transparent about this leakage, we train SciNCL on two datasets:

SPECTER replication (w/ leakage): We replicate SPECTER's training data including its leakage. Unfortunately, SPECTER provides neither citation data nor a mapping to S2ORC, which our citation embeddings are based on. We successfully map 96.2% of SPECTER's query papers and 83.3% of the corpus from which positives and negatives are sampled to S2ORC. To account for the missing papers, we randomly sample papers from S2ORC (without the SciDocs papers) such that the absolute number of papers is identical with SPECTER.

S2ORC subset (w/o leakage): We select a random subset from S2ORC that does not contain any of the mapped SciDocs papers. This avoids SPECTER's leakage, but also makes the scores reported in Cohan et al. (2020) less comparable. We successfully map 98.6% of the SciDocs papers to S2ORC. Thus, only the remaining 1.4% of the SciDocs papers could leak into this training set.

The details of the dataset creation are described in Appendix A.2 and A.4. Both training sets yield 684K triplets (same count as SPECTER). Also, the ratio of training triplets per query remains the same (§3.4). Our citation embedding model is trained on the S2ORC citation graph. In *w/ leakage*, we include all SPECTER papers even if they are part of SciDocs, the remaining SciDocs papers are excluded (52.5 nodes and 463M edges). In *w/o leakage*, all mapped SciDocs papers are excluded (52.4M nodes and 447M edges) such that we avoid leakage also for the citation embedding model.

4.3 Model 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 computing 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). The graph embedding training is performed on an Intel Xeon Gold 6230 CPU with 60 cores and takes approx. 6 hours. The kNN strategy is implemented with FAISS (Johnson et al., 2021) using a flat index (exhaustive search) and takes less than 30min for indexing and retrieval of the triplets.

4.4 Baseline Methods

We compare against the following baselines (details in Appendix A.6): USE (Cer et al., 2018), BERT (Devlin et al., 2019), BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019), CiteBERT (Wright and Augenstein, 2021), DeCLUTR (Giorgi et al.,

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	
Oracle SciDocs †	*87.1*	*94.8*	*87.2*	*93.5*	*88.7*	*94.6*	*92.3*	*96.8*	*91.4*	*96.4*	*53.8*	*19.4*	*83.0*
USE (2018)	80.0	83.9	77.2	88.1	76.5	88.1	76.6	89.0	78.3	89.8	53.7	19.6	75.1
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
DeCLUTR (2021)	81.2	88.0	63.4	80.6	60.0	78.6	57.2	77.4	62.9	80.9	52.0	17.4	66.6
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
Replicated SPECTER training data (w/ leakage):													
SciNCL (ours)	81.4	**88.7**	**85.3**	**92.3**	**87.5**	**93.9**	**93.6**	**97.3**	**91.6**	**96.4**	**53.9**	19.3	**81.8**
± σ w/ ten seeds	*.449*	*.422*	*.128*	*.08*	*.162*	*.118*	*.104*	*.054*	*.099*	*.066*	*.203*	*.356*	*.064*
Random S2ORC training data (w/o leakage):													
SPECTER	81.3	88.4	83.1	91.3	84.0	92.1	86.2	93.9	87.8	94.7	52.2	17.5	79.4
SciNCL (ours)	81.3	89.4	84.3	91.8	85.6	92.8	91.4	96.3	90.1	95.7	54.3	19.9	81.1

Table 1: Results on the SciDocs test set. With replicated SPECTER training data, SciNCL 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. With training data randomly sampled from S2ORC, SciNCL outperforms SPECTER in terms of avg. score with 1.7 points. The scores with * are from Cohan et al. (2020). *Oracle SciDocs* † is the upper bound of the performance with triplets from SciDocs's data.

2021), the graph-convolution approach SGC (Wu et al., 2019), Citeomatic (Bhagavatula et al., 2018), and SPECTER (Cohan et al., 2020).

Also, we compare against *Oracle SciDocs* which is identical to SciNCL except that its triplets are generated based on SciDocs's validation and test set using their gold labels. For example, papers with the same MAG labels are positives and papers with different labels are negatives. In total, this procedure creates 106K training triplets for *Oracle SciDocs*. Accordingly, *Oracle SciDocs* represents an estimate for the performance upper bound that can be achieved with the current setting (triplet margin loss and SciBERT encoder).

5 Overall Results

Tab. 1 shows the results, comparing SciNCL with the best validation performance against the baselines. With replicated SPECTER training data (w/ leakage), SciNCL achieves an average performance of 81.8 across all metrics, which is a 1.8 point absolute improvement over SPECTER (the next-best baseline). When trained without leakage, the improvement of SciNCL over SPECTER is consistent with 1.7 points but generally lower (79.4 avg. score). In the following, we refer to the results obtained through training on the replicated SPECTER data (w/ leakage) if not otherwise mentioned.

We find the best validation performance based on SPECTER's data 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. SciNCL's scores are reported as mean over ten random seeds (seed $\in [0, 9]$).

For MAG classification, SPECTER achieves the best result with 82.0 F1 followed by SciNCL with 81.4 F1 (-0.6 points). For MeSH classification, SciNCL yields the highest score with 88.7 F1 (+2.3 compared to SPECTER). Both classification tasks have in common that the chosen training settings lead to over-fitting. Changing the training by using only 1% training data, SciNCL yields 82.2 F1@MAG (Tab. 2). In all user activity and citation tasks, SciNCL yields higher scores than all baselines. Moreover, SciNCL outperforms SGC on direct citation prediction, where SGC outperforms SPECTER in terms of nDCG. On the recommender task, SPECTER yields the best P@1 with 20.0, whereas SciNCL achieves 19.3 P@1 (in terms of nDCG SciNCL and SPECTER are on par).

When training SPECTER and SciNCL without leakage, SciNCL outperforms SPECTER even in 11 of 12 metrics and is on par in the other metric. This suggests that SciNCL's hyperparameters have a low corpus dependency since they were only optimized on the corpus with leakage.

Regarding the LLM baselines, we observe that the general-domain BERT, with a score of 63.4, outperforms the domain-specific BERT variants, namely SciBERT (59.6) and BioBERT (58.8). LLMs without citations or contrastive objectives yield generally poor results. This emphasizes the anisotropy problem of embeddings directly extracted from current LLMs and highlights the advantage of combining text and citation information.

In summary, we show that SciNCL's triplet selection leads on average to a performance improvement on SciDocs, with most gains being observed for user activity and citation tasks. The gain from 80.0 to 81.8 is particularly notable given that even *Oracle SciDocs* yields with 83.0 an only marginally higher avg. score despite using test and validation data from SciDocs for the triplet selection.

6 Impact of Sample Difficulty

In this section, we present the optimization of SciNCL's sampling strategy (§3.3). We optimize the sampling for positives and hard or easy negatives with partial grid search on a random sample of 10% of the replicated SPECTER training data (sampling based on queries). Our experiments show that optimizations on this subset correlate with the entire dataset. The validation scores in Fig. 2 and 3 are reported as the mean over three random seeds.

6.1 Positive Samples

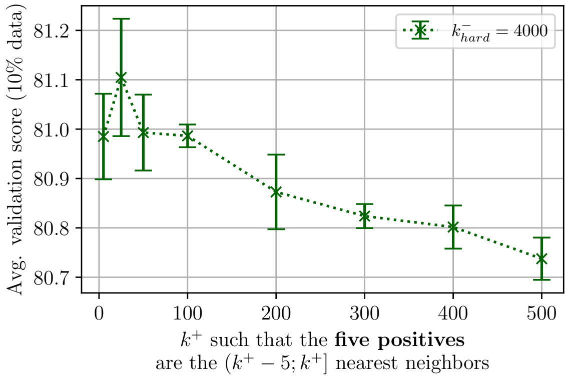


Figure 2: Results on the validation set w.r.t. positive sampling with kNN when using 10% training data.

Fig. 2 shows the avg. 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 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).

6.2 Hard Negative Samples

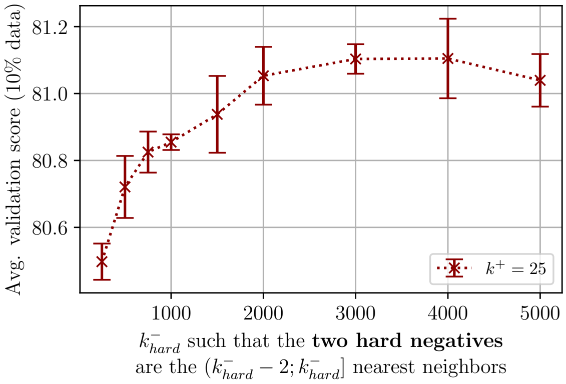


Figure 3: Results on the validation set w.r.t. hard negative sampling with kNN using 10% training data.

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

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 difference is marginal. When rounded to one decimal, their average test scores are identical. The

	CLS	USR	CITE	REC	Avg.	Δ
SciNCL	85.0	88.8	**94.7**	36.6	**81.8**	–
SPECTER	84.2	88.4	91.5	36.9	80.0	-1.8
k^{-}_{hard}=2000	84.9	88.8	**94.7**	36.1	81.6	-0.2
k^{-}_{hard}=3000	84.5	88.7	94.6	36.9	81.7	-0.1
easy neg. w/ random	85.1	88.8	**94.7**	36.6	**81.8**	0.0
Init. w/ BERT-Base	83.4	88.4	93.8	37.5	81.2	-0.6
Init. w/ BERT-Large	84.6	88.7	94.1	36.4	81.4	-0.4
Init. w/ BioBERT	83.7	88.6	93.8	**37.7**	81.4	-0.4
1% training data	85.2	88.3	92.7	36.1	80.8	-1.0
10% training data	85.1	88.7	93.5	36.2	81.1	-0.6
BitFit training	**85.8**	88.6	93.7	35.3	81.2	-0.5

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

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.

7 Ablation Analysis

Next, we evaluate the impact of language model initialization and number of parameters and triples.

7.1 Initial Language Models

Tab. 2 shows the effect of initializing the model weights not with SciBERT but with general-domain LLMs (BERT-Base and BERT-Large) or with BioBERT. The initialization with other LLMs decreases the performance. However, the decline is marginal (BERT-Base -0.6, BERT-Large -0.4, BioBERT -0.4) and all LLMs outperform the SPECTER baseline. For the recommendation task, in which SPECTER is superior over SciNCL, BioBERT outperforms SPECTER. This indicates that the improved triplet mining of SciNCL has a greater domain adaption effect than pretraining on domain-specific literature. Given that pretraining of LLMs 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 Computing Efficiency

The last three rows of Tab. 2 show the results regarding data and computing efficiency. When keeping the citation graph unchanged but training the language model with only 10% of the original triplets, SciNCL still yields a score of 81.1 (-0.6). Even with only 1% (6840 triplets), 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 *textual* sample efficiency, one could manually create triplets or use existing supervised datasets as in Gao et al. (2021).

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 81.2 (-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 LLM and data setups can be reused. This improvement over SPECTER can be also observed when excluding the SciDocs papers during training (see w/o leakage in Tab. 1). Furthermore, SciNCL's improvement from 80.0 to 81.8 is particularly notable given that even *oracle triplets*, which are generated with SciDocs's test and validation data, yield with 83.0 only a marginally higher score.

Our work highlights the importance of sample generation in a contrastive learning setting. We show that language model training with 1% of triplets is sufficient to outperform SPECTER, whereas the remaining 99% provide only 1.0 additional points (80.8 to 81.8). This sample efficiency is achieved by adding reasonable effort for sample generation, i.e., graph embedding training and kNN search. We also demonstrate that in-domain LLM pretraining (like SciBERT) is beneficial, while general-domain LLMs can achieve 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 an LLM from scratch is infeasible.

References

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.

Robin Brochier, Adrien Guille, and Julien Velcin. 2019. Global Vectors for Node Representations. In *The World Wide Web Conference on - WWW '19*, volume 2, pages 2587–2593, New York, New York, USA. ACM Press.

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.

Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *arXiv:1803.11175*.

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.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. CERT: Contrastive Self-supervised Learning for Language Understanding. *arXiv:2005.12766*, pages 1–16.

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.

Bela Gipp and J Beel. 2009. Citation Proximity Analysis (CPA) - A new approach for identifying related work based on Co-Citation Analysis. *Birger Larsen and Jacqueline Leta, editors, Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09)*, 2(July):571–575.

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.

Andreas Nugaard Holm, Barbara Plank, Dustin Wright, and Isabelle Augenstein. 2022. Longitudinal Citation Prediction using Temporal Graph Neural Networks. In *AAAI 2022 Workshop on Scientific Document Understanding (SDU 2022)*.

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.

Parminder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. 2021. Comparative analysis on cross-modal information retrieval: A review. *Computer Science Review*, 39:100336.

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.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence 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 2528–2540, Online. Association for Computational Linguistics.

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.

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.

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.

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.

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. 2021a. A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned and Perspectives. *arXiv:2102.12982*.

Nils Rethmeier and Isabelle Augenstein. 2021b. Data-Efficient Pretraining via Contrastive Self-Supervision. *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.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

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, Dilip Krishnan, and Phillip Isola. 2020a. Contrastive representation distillation. In *International Conference on Learning Representations*.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020b. 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.

X. Wang, Yufei Ye, and Abhinav Kumar Gupta. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6857–6866.

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 the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1796–1807, Online. 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.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. Smoothed Contrastive Learning for Unsupervised Sentence Embedding. *arXiv:2109.04321*.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*, pages 1–16.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network representation learning with rich text information. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 2111–2117. AAAI Press.

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 Extended Related Work

Triplet Mining remains a challenge in NLP due to the discrete nature of language which makes data augmentation less trivial as compared to computer vision (Gao et al., 2021). Examples for augmentation strategies are: translation (Fang et al., 2020), or word deletion and reordering (Wu et al., 2020). Positives and negatives can be sampled based on the sentence position within a document (Giorgi et al., 2021). Gao et al. (2021) utilize supervised entailment datasets for the triplet generation.

Language- and text-independent approaches are also applied: Kim et al. (2021) use intermediate BERT hidden state for positive sampling and Wu et al. (2021) add noise to representations to obtain negative samples. Xiong et al. (2020) present an approach similar to SciNCL where they sample hard negatives from the k nearest neighbors in the embedding space derived from the previous model checkpoint. While Xiong et al. rely only on textual data, SciNCL integrates also citation information which are especially valuable in the scientific context as Cohan et al. (2020) have shown.

A.2 Mapping to S2ORC

Table 3: Mapping to S2ORC citation graph

S2ORC mapping	Success rate
SciDocs papers	
- with S2ORC IDs	220,815 / 223,932 (98.6%)
- in S2ORC graph	197,811 / 223,932 (88.3%)
SPECTER papers	
- with S2ORC IDs	311,094 / 311,860 (99.7%)
- in S2ORC graph	260,014 / 311,860 (83.3%)

Neither the SPECTER training data nor the SciDocs test data comes with a mapping to the S2ORC dataset, which we use for the training of the citation embedding model. However, to replicate SPECTER's training data and to avoid leakage of SciDocs test data such a mapping is needed. Therefore, we try to map the papers to S2ORC based on PDF hashes and exact title matches. The remaining paper metadata is collected through the Semantic Scholar API. Tab. 3 summarizes the outcome of mapping procedure. Failed mappings can be attributed to papers being unavailable through the Semantic Scholar API (e.g., retracted papers) or papers not being part of S2ORC citation graph.

A.3 SPECTER-SciDocs Leakage

When replicating SPECTER (Cohan et al., 2020), we found a substantial overlap between the papers[3] used during the model training and the papers from their SciDocs benchmark[4]. In both datasets, papers are associated with Semantic Scholar IDs. Thus, no custom ID mapping as in App. A.2 is required to identify papers that leak from training to test data. From the 311,860 unique papers used in SPECTER's training data, we find 79,201 papers (25.4%) in the test set of SciDocs and 79,609 papers (25.5%) in its validation set. When combining test and validation set, there is a total overlap of 126,176 papers (40.5%). However, this overlap affects only the 'unsupervised' paper metadata (title, abstract, citations, etc.) and not the gold labels used in SciDocs (e.g., MAG labels or clicked recommendations).

A.4 Dataset Creation

As describe in §4.2, we conduct our experiments on two datasets. Both datasets rely on the citation graph of S2ORC (Lo et al., 2020). More specifically, S2ORC with the version identifier `20200705v1` is used. The full citation graph consists of 52.6M nodes (papers) and 467M edges (citations). Tab. 4 presents statistics on the datasets and their overlap with SPECTER and SciDocs. The steps to reproduce both datasets are:

Replicated SPECTER (w/ leakage) In order to replicate SPECTER's training data and do not increase the leakage, we exclude all SciDocs papers which are not used by SPECTER from the S2ORC citation graph. This means that apart from the 110,538 SPECTER papers not a single other SciDocs paper is included. The resulting citation graph has 52.5M nodes and 463M edges and is used for training the citation graph embeddings.

For the SciNCL triplet selection, we also replicate SPECTER's query papers and its corpus from which positive and negatives are sampled. Our mapping and the underlying citation graph allows us to use 227,869 of 248,007 SPECTER's papers for training. Regarding query papers, we use 131,644 of 136,820 SPECTER's query papers. To align the number training triplets with the one from SPECTER, additional papers are randomly sampled from the filtered citation graph.

[3] `https://github.com/allenai/specter/issues/2`

[4] `https://github.com/allenai/scidocs`

Random S2ORC subset (w/o leakage) To avoid leakage, we exclude all successfully mapped SciDocs papers from the S2ORC citation graph. After filtering the graph has 52.3 nodes and 447M edges. The citation graph embedding model is trained on this graph.

Next, we reproduce triplet selection from SPECTER. Any random 136,820 query papers are selected from the filtered graph. For each query, we generate five positives (cited by the query), two hard negatives (citation of citation), and three random nodes from the filtered S2ORC citation graphs. This sampling produces 684,100 training triplets with 680,967 unique papers IDs (more compared to the replicated SPECTER dataset). Based on these triplets the SPECTER model for this dataset is trained with the same model settings and hyperparameters as SciNCL (second last row in Tab. 1).

Lastly, the SciNCL triplets are generated based on the citation graph embeddings of the same 680,967 unique papers IDs, i.e, the FAISS index contains only these papers and not the remaining S2ORC papers. Also, the same 136,820 query papers are used.

Table 4: Statistics for our two datasets and their overlap with SPECTER and SciDocs respectively.

	Replicated SPECTER (w/ leakage)	**Random S2ORC subset (w/o leakage)**
Training triplets	684,100	684,100
Unique paper IDs	248,007	680,967
- in SPECTER	227,869	9,182
- in SciDocs	110,538	0
- in SciDocs and in SPECTER	110,538	0
Query paper IDs	136,820	136,820
- in SciDocs	69,306	0
- in SPECTER queries	131,644	463
Citation graph		
- Nodes	52,526,134	52,373,977
- Edges	463,697,639	447,697,727

A.5 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 λ=0.1 (based on the recommended settings by Lerer et al. (2019)).

Table 5: 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. 5 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.6 Baseline Details

If not otherwise mentioned, all BERT variations are used in their *base-uncased* versions.

The weights for BERT (*bert-base-uncased*), BioBERT (*biobert-base-cased-v1.2*), CiteBERT (*citebert*), DeCLUTR (*declutr-sci-base*) are taken from Huggingface Hub[5]. We use Universal Sentence Encoder (USE) from Tensorflow Hub[6]. For *Oracle SciDocs*, we use the SciNCL implementation and under-sample the triplets from the classification tasks to ensure a balanced triplet distribution over the tasks. The SPECTER version for the random S2ORC training data (w/o leakage) is also trained with the SciNCL implementation. Please see Cohan et al. (2020) for additional baseline methods and their implementation details.

A.7 Negative Results

We investigated additional sampling strategies and model modification of which none led to a significant performance improvement.

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

[6] https://tfhub.dev/google/universal-sentence-encoder-large/5

A.7.1 Undirected Citations

Our graph embedding model considers citations as directed edges by default. We also train a SciNCL model with undirected citations by first converting a single edge (a, b) into the two edges (a, b) and (b, a). This approach yields a slightly worse performance (81.7 avg. score; -0.1 points) and, therefore, was discarded for the final experiments.

A.7.2 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.

A.7.3 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. However, this decreased the performance.

A.7.4 Sampling with Similarity Threshold

As alternative to kNN, we select samples based on cosine similarity in the citation embedding space. 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 larger than the threshold would be 0.8 and 0.7. The corresponding papers would be selected as samples. While the positive threshold t^+ should close to 1, the negative threshold t^- should be small to ensure samples are dissimilar from d^Q. However, the empirical results suggest that this strategy is inferior compared to kNN.

A.7.5 Hard Negatives with Similarity Threshold

Selecting hard negatives based on the similarity threshold yields a test score of 81.7 (-0.1 points).

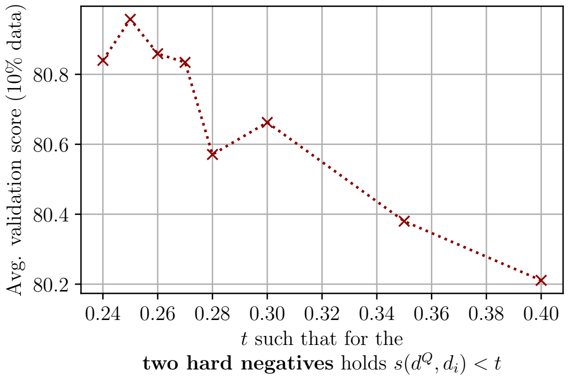


Figure 4: Results on the validation set w.r.t. hard negative sampling with Sim using 10% training data.

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.

A.7.6 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.

A.7.7 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.

A.7.8 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).

A.7.9 Student-Teacher Learning

Student-teacher learning is effective in related work on cross-modal knowledge transfer (Kaur et al., 2021; Tian et al., 2020a). We also try to adopt this approach for our experiments, whereby the Transformer language model is the student, and the citation graph embedding model is the teacher. By directly learning from the citation embeddings, we could circumvent the positive and negative sampling needed for triplet loss learning, which introduces unwanted issues like collisions. Given a batch of document representations derived from text D_{Text} (through the language model) and the citation graph representations for the same documents D_{Graph}, we compute the pairwise cosine similarity for both sets S_{Text} and S_{Graph}. To transfer the knowledge from the citation embeddings into the language model, we devise the student-teacher loss $\mathcal{L}_{ST}$ based on a mean-squared-error loss (MSE) such that the difference between the cosine similarities is minimized:

$$\mathcal{L}_{ST} = \text{MSE}(S_{Text}, S_{Graph}) \qquad (1)$$

Despite the promising results from Tian et al. (2020a), the student-teacher approach performs poorly in our experiments. We attribute this the overfitting to the citation data (the training loss approaches zero after a few steps while the validation loss remains high). The model trained with $\mathcal{L}_{ST}$ yields only a SciDocs average score of 64.7, slightly better than SciBERT but substantially worse than SciNCL with triplet loss.

Additionally, we experiment with a joint loss that is the sum of triplet margin loss $\mathcal{L}_{Triplet}$ (see §3.1) and the student-teacher loss $\mathcal{L}_{ST}$:

$$\mathcal{L}_{Joint} = \mathcal{L}_{Triplet} + \mathcal{L}_{ST} \qquad (2)$$

Training with the joint loss $\mathcal{L}_{Joint}$ achieves an average score of 80.5. Even though the joint loss is not subject to overfitting, its SciDocs performance is slightly worse than the triplet loss $\mathcal{L}_{Triplet}$ alone. Given this outcome and that the computation of the cosine similarities adds additional complexity, we discard the student-teacher approach for the final experiments.

A.7.10 SPECTER & Bidirectional Citations

SPECTER (Cohan et al., 2020) relies on unidirectional citations for their sampling strategy. While papers *cited by* the query paper are considered as positives samples, those *citing* the query paper (opposite citation direction) could be negative samples. We see this use of citations as a conceptional flaw in their sampling strategy.

To test the actual effect on the resulting document representation, we first replicate the original unidirectional sampling strategy from SPECTER with our training data (see w/ leakage in §4.2). The resulting SPECTER model achieves an average score of 79.0 on SciDocs.[7] When changing the sampling strategy from unidirectional to bidirectional ('citations to the query' are also treated as a signal for similarity), we observe an improvement of +0.4 points to 79.4. Consequently, the use of unidirectional citations is not only a conceptional issue but also degrades learning performance.

A.8 Collisions

Similar to SPECTER, SciNCL's sampling based on graph embeddings could cause collisions when selecting positives and negatives from regions close to each other. To avoid this, we rely on a sample induced margin that is defined by the hyperparameter k^+ and k^-_{hard} (distance between red and green band in Fig. 1). When the margin gets too small, positives and negatives are more likely to collide. A collision occurs when the paper pair (d_q, d_s) is contained in the training data as positive and as negative sample at the same time.

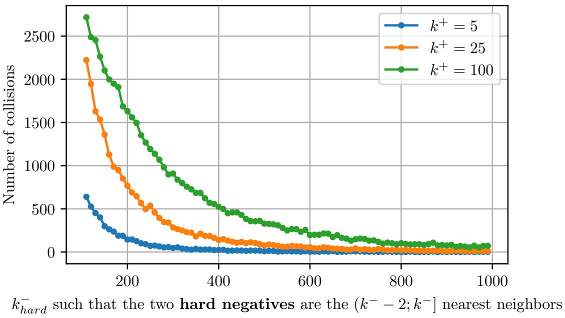


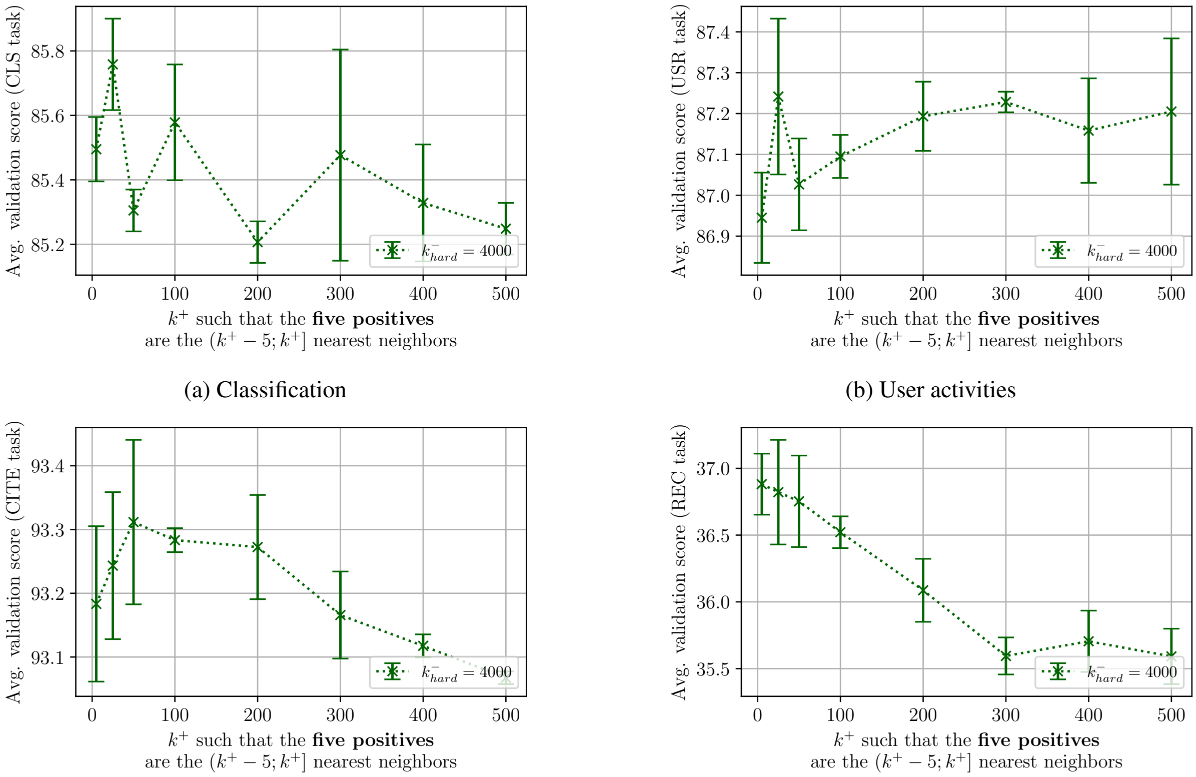
Figure 5: Number of collisions w.r.t. size of the sample induced margin as defined through k^+ and k^-_{hard}.

Fig. 5 demonstrates the relation between the number of collisions and the size of the sample induced margin. The number of collisions increases when the sample induced margin gets smaller. The opposite is the case when the margin is large enough ($k^-_{\text{hard}} > 1000$), i.e., then the number of collisions goes to zero. This relation also affects the evaluation performance as Fig. 2 and Fig. 3 show. Namely, for large k^+ or small k^-_{hard} SciNCL's performance declines and approaches SPECTER's performance.

A.9 Task-specific Results

Fig. 6 and 7 present the validation performance like in §6 but on a task-level and not as an average over

[7] The difference to the scores reported in Cohan et al. (2020) is due to the difference in the underlying training data.



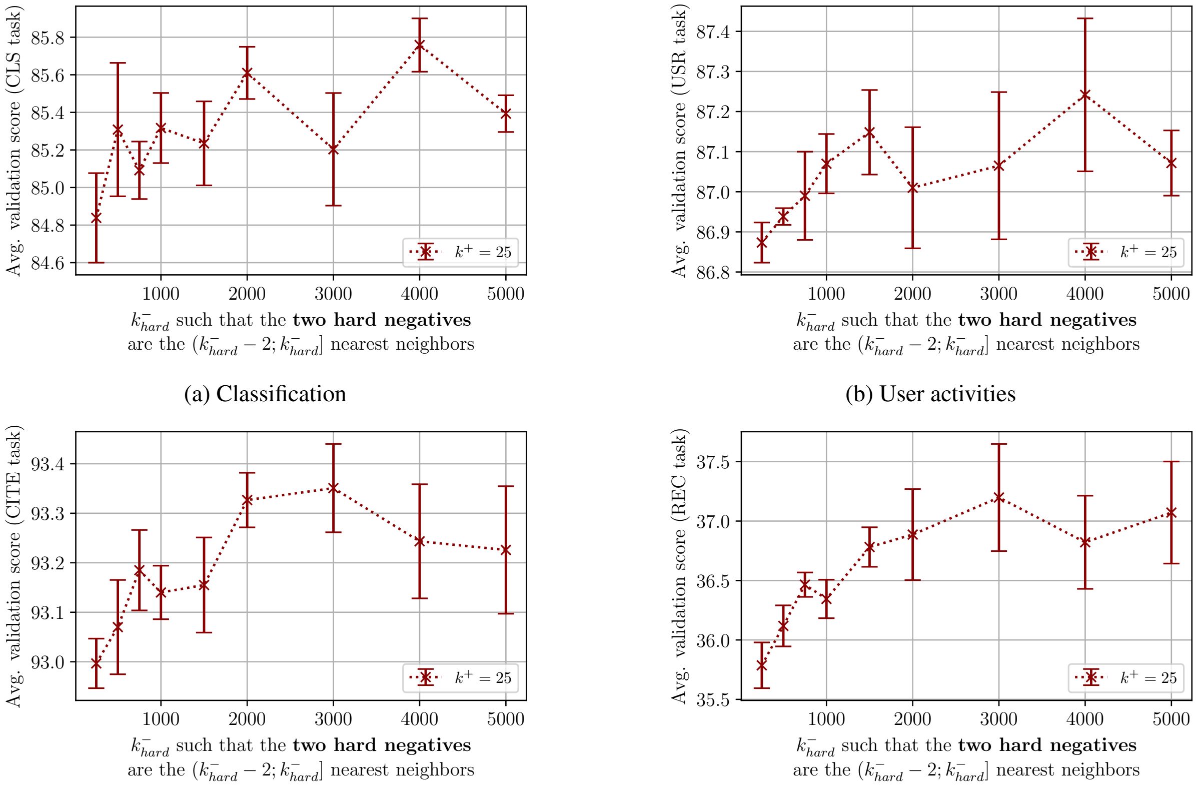
(a) Classification

(b) User activities

(c) Citation

(d) Recommendation

Figure 6: 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 7: Task-level validation performance w.r.t. k^-_{hard} with KNN strategy using 10% training data.

all tasks. The plots show that the optimal k^+ and k^-_{hard} values are partially task dependent.

A.10 Examples

Tab. 6 lists three examples of query papers with their corresponding positive and negative samples. The complete set of triplets that we use during training is available in our code repository[1].

Table 6: 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