

Large-scale Evaluation of Transformer-based Article Encoders on the Task of Citation Recommendation

Zoran Medić and Jan Šnajder

Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{zoran.medic, jan.snajder}@fer.hr

Abstract

Recently introduced transformer-based article encoders (TAEs) designed to produce similar vector representations for mutually related scientific articles have demonstrated strong performance on benchmark datasets for scientific article recommendation. However, the existing benchmark datasets are predominantly focused on single domains and, in some cases, contain easy negatives in small candidate pools. Evaluating representations on such benchmarks might obscure the realistic performance of TAEs in setups with thousands of articles in candidate pools. In this work, we evaluate TAEs on large benchmarks with more challenging candidate pools. We compare the performance of TAEs with a lexical retrieval baseline model BM25 on the task of citation recommendation, where the model produces a list of recommendations for citing in a given input article. We find out that BM25 is still very competitive with the state-of-the-art neural retrievers, a finding which is surprising given the strong performance of TAEs on small benchmarks. As a remedy for the limitations of the existing benchmarks, we propose a new benchmark dataset for evaluating scientific article representations: **Multi-Domain Citation Recommendation dataset (MDCR)**, which covers different scientific fields and contains challenging candidate pools.

1 Introduction

The introduction of large pre-trained language models (LMs) (Devlin et al., 2019; Radford et al., 2019; Lewis et al., 2020; Raffel et al., 2020) based on the transformer architecture (Vaswani et al., 2017) has improved performance on numerous NLP tasks. The adaptation of LMs to scientific corpora (Beltagy et al., 2019; Luu et al., 2021; Gupta et al., 2022; Lee et al., 2020) laid the foundation for applying transformer-based LMs to various scholarly document processing (SDP) tasks, such as named-entity recognition (Naseem et al., 2020), article summa-

rization (Cai et al., 2022), scientific fact-checking (Wadden et al., 2020), describing relationships between articles (Luu et al., 2021), and citation recommendation (CR) (Nogueira et al., 2020; Gu et al., 2022), among others.

While some of the SDP tasks rely on word- or sentence-level representations, others, such as CR and article summarization, require document-level representations. To obtain such representations, recent work has proposed various transformer-based article encoders (TAEs), i.e., LMs that are finetuned using citation or co-citation information as a training signal, such as SPECTER (Cohan et al., 2020), ASPIRE (Mysore et al., 2021a), and SciNCL (Ostendorff et al., 2022). Representations obtained with these models can then be used in various downstream recommendation tasks where a user searches for articles that are in some way relevant to a given query article.

To date, article representations obtained with TAEs have been evaluated against recommendation benchmarks such as SCIDOCs (Cohan et al., 2020), RELISH (Brown et al., 2019) or TRECCOVID (Voorhees et al., 2021). While SCIDOCs focuses mainly on the field of computer science, RELISH and TRECCOVID cover articles from the biomedical field. These benchmarks contain a set of query articles, where each query is paired with a candidate pool consisting of both relevant and irrelevant articles for that query. The difference between the benchmarks, apart from the domains they cover, is how the candidate pools are constructed: RELISH and TRECCOVID contain expert-annotated relevance labels for each candidate in a pool, while SCIDOCs uses random sampling of negative candidates for pool construction. However, they all contain relatively small candidate pools (e.g., 25 in SCIDOCs). Such small and, in some cases, randomly sampled candidate pools do not resemble typical use-case scenarios in which query articles are compared to millions of candidate articles from

public databases. Thus, evaluating TAEs on such benchmarks can lead to an overly optimistic performance estimation as the candidate pool does not constitute a representative sample of the population of candidate articles in realistic use cases.

In this work, we turn to a more realistic evaluation of TAEs and evaluate them on large ($\geq 200k$) candidate pools and across different scientific fields. Although emulating such a realistic setup has so far been avoided due to the prohibitive computational cost of nearest neighbor search on millions of embeddings, research on GPU-based nearest neighbor (NN) search (Johnson et al., 2019) has given rise to efficient techniques that enable embedding-based search in large-scale setups. To make use of fast NN search, we focus on the bi-encoder models (Lin et al., 2021), that can be easily coupled with fast GPU-based NN search. We evaluate TAEs on the task of CR, in which a model outputs a list of articles as recommendations for citing in a given article. Alongside TAEs, we evaluate the traditional lexical retrieval model BM25 (Robertson and Walker, 1994), which, in spite of its simplicity, still stands as a hard-to-beat baseline in many retrieval tasks. Our evaluation shows that BM25 performs on par with TAEs in this setup, especially as candidate pools grow.

Building on the results of our large-scale evaluation of TAEs, we then construct a new benchmark dataset for evaluating scientific article representations on the task of CR. Our Multi-Domain CR-based benchmark dataset (MDCR), albeit comparable in size to previous benchmarks, spans different scientific fields and consists of challenging candidate pools. More precisely, candidate pools in MDCR contain different candidate types, ranging from those obtained from the large-scale evaluation of state-of-the-art TAEs to candidates from the citation graph neighborhood.

To summarize, the contribution of our work is twofold: (1) we conduct a large-scale evaluation of state-of-the-art TAEs on pools of varying sizes, and (2) present a new and challenging multi-domain benchmark dataset for evaluating scientific article representations that contains challenging candidates identified in the large-scale evaluation.¹

The rest of the paper is organized as follows. In Section 2, we describe the models we evaluate and give an overview of the existing benchmarks

for scientific article recommendation. Section 3 presents the results of a large-scale evaluation of TAEs and BM25 in two evaluation setups. In Section 4 we describe the construction of a new and more challenging multi-domain benchmark and present the initial results for the models we considered. Section 5 concludes the paper and proposes future work.

2 Models and Benchmarks

2.1 Transformer-based Article Encoders

As a baseline TAE, we consider **SCIBERT** (Beltagy et al., 2019), a variant of BERT (Devlin et al., 2019), trained on a corpus of scientific articles with masked language modeling objective. Next, we include **SPECTER** (Cohan et al., 2020), a SCIBERT-based TAE trained with a contrastive learning objective that minimizes the L2 distance between embeddings of citing-cited article pairs. Further, we consider **SCINCL** (Ostendorff et al., 2022), another SCIBERT-based TAE that uses citation graph embeddings for a more informative selection of negative examples with the same contrastive learning objective as SPECTER. Finally, we also evaluate **ASPIRE** (Mysore et al., 2021a), a TAE that uses a co-citation signal to make sentence embeddings of co-cited articles similar.

Among these four TAEs, only SCIBERT is trained without any inter-article (i.e., citation or co-citation) training signal. We thus consider it as a baseline to investigate how well LMs pre-trained on domain’s corpora can be used in retrieval scenarios without any finetuning. On the other hand, the rest of the TAEs differ both in the type of inter-article training signal used (co-citation for ASPIRE vs. citation for SPECTER and SCINCL) and in the granularity of representation used for article matching (sentence embeddings for ASPIRE vs. document embeddings for SPECTER and SCINCL). All the considered TAEs are *bi-encoders* (Lin et al., 2021), i.e., they produce dense representations of a single input article, which allows them to be easily employed in large-scale setups when coupled with fast nearest neighbor search methods. The alternative are the *cross-encoders*, which take two concatenated articles as the input and output the relevance matching score. Although the cross-encoders often outperform bi-encoders, we do not consider them here as they are not compatible with nearest neighbor search methods and therefore not suitable for

¹Our code, the data splits and the new benchmark data are publicly available at the following link: <https://github.com/zoranmedic/mdcr>.

large-scale retrieval.²

All the considered TAEs produce scientific article representations using the article’s title and abstract as input. Since the title and abstract serve as a condensed overview of an article, it is clear that not all possible relationships between a pair of articles can be detected using such input only. However, we consider the title and abstract a reliable proxy for otherwise complex and computationally expensive processing of the whole article’s content.

2.2 Existing Benchmarks

Scientific article recommendation benchmarks that TAEs were evaluated on so far were designed for domain-specific retrieval evaluation across small-sized and, in some cases, randomly sampled candidate pools. Each benchmark consists of a set of queries, where each query (title and abstract or a free-form text) is paired with a corresponding *candidate pool*, i.e., a set of query-relevant (*positive*) and query-irrelevant (*negative candidates*) articles. We review the most commonly used benchmarks below.

SCIDOCS (Cohan et al., 2020): A collection of datasets for the evaluation of classification and retrieval tasks that use abstract-level article representations. In retrieval tasks, each query article is paired with a candidate pool of 5 positive and 25 randomly sampled negative candidates.

RELISH (Brown et al., 2019): A collection of query and candidate articles expert-annotated for relevance. Query articles are from the field of biomedicine, each paired with a set of 60 candidates.

TRECCOVID (Voorhees et al., 2021): A TREC-style benchmark consisting of various queries related to COVID-19. Each query is paired with around 300 candidate articles annotated for relevance by medical experts.

CSFCUBE (Mysore et al., 2021b): An expert-annotated dataset of 50 computer science articles annotated at sentence-level for aspect-based relevance with candidate articles. The average candidate pool size is 125.

Three of these benchmarks (RELISH, TRECCOVID, CSFCUBE) are single-domain by design, while SCIDOCS is constructed with queries from different scientific fields. However, the majority of SCIDOCS queries (over 70%) come from a single domain (computer science), making it a predominantly computer science-oriented benchmark.

Existing benchmarks also differ in how the candidate pools in each of them were constructed. While RELISH, TRECCOVID, and CSFCUBE contain expert-annotated candidate pools, meaning that field experts annotated the relevance of each candidate to the query, candidate pools for retrieval tasks in SCIDOCS are made of negative candidates randomly sampled from a set of articles that are not related to the query. For example, in the case of the “Cite” task in SCIDOCS, each query article is paired with a pool of 5 articles cited in the query, and 25 negative candidates are randomly sampled from a held-out set of articles not cited in the query article. An obvious advantage of random candidate pools over expert-annotated pools is that they are less expensive to construct. However, a downside is that random candidate pools might contain many candidates that are entirely unrelated to the query and lead to overly optimistic performance estimates that are not representative of realistic large-scale retrieval scenarios.

3 Large-Scale Evaluation

We performed the large-scale evaluation in two setups: dataset- and field-level. Dataset-level evaluation resembles a basic evaluation setup – a random sampling of both queries and articles in the candidate pool. The field-level evaluation focuses on specific scientific fields using queries and candidate pools comprised of articles from specific fields.

In both setups, we evaluated the chosen models on the task of global CR, in which a model is trained to produce a list of articles as recommendations for citing in a given query article. Although CR is not the only task on which TAEs can be evaluated, it is arguably the most accessible among the article retrieving tasks. Whereas other tasks (e.g., user activity tasks) might require data that is typically not publicly available (e.g., search engine logs), CR datasets are easily obtained through parsing reference lists of publicly available articles. Previous research on global CR has proposed many features that could be used to represent the input articles (Bhagavatula et al., 2018; Ali et al., 2021).

²We thus only consider TS-ASPIRE model in our work and leave out OT-ASPIRE, a variant that uses optimal transport over sentence embeddings, whose computational complexity prohibits its use in large-scale retrieval scenarios.

However, in this work, we only use the article’s title and abstract as input, as our focus is not on improving the state-of-the-art in global CR but rather on evaluating the TAE-produced article representations in a retrieval scenario. For a detailed overview of the various tasks and methods in CR, we refer the reader to (Medić and Šnajder, 2020).

For each TAE that we consider, the input was constructed by concatenating the input article’s title and abstract (separated with a [SEP] token). For SCIBERT, SPECTER, and SCINCL, we used the final layer’s [CLS] token embedding as input article’s representation, while for ASPIRE we mean-pooled token embeddings across all layers for each sentence in the input. We used HuggingFace’s³ implementations of TAEs, while for BM25 we used Lucene’s implementation, i.e., its Python toolkit `pyserini`.⁴ For nearest neighbor search across article embeddings, we used Faiss (Johnson et al., 2019).⁵

We used the S2ORC dataset (Lo et al., 2020) in all our experiments. S2ORC is a recently released large dataset of 81.1M scientific articles covering dozens of scientific fields. Together with the metadata and article’s title and abstract, the dataset contains citation links between the articles. Therefore, we consider it appropriate for the large-scale evaluation, not just due to its size and coverage but recency as well. We perform initial filtering of articles and remove all those with (1) empty publication year field, (2) empty title field, (3) abstract shorter than 30 characters, or (4) less than three citations in S2ORC. This filtering leaves us with a *prefiltered set* of around 16M articles that we use for both sampling of queries and candidate pool construction in both evaluation setups.

For both evaluation setups, we report the standard metrics used in prior work on scientific article recommendation: MAP, NDCG, and R@30. We set k in R@ k to 30, since on average there are 29 positives (cited articles) for each query in the query set. All metrics range from 0 to 1, where higher is better. Although defined differently, all the metrics yield higher values when relevant articles are positioned higher in the list of retrieved articles.

³<https://github.com/huggingface/transformers>

⁴<https://github.com/castorini/pyserini>

⁵<https://github.com/facebookresearch/faiss>

3.1 Dataset-level

We start by describing the dataset-level setup in which we evaluated how TAEs perform when asked to provide recommendations over a large candidate pool for random queries from S2ORC.

First, we sampled a random set of 3800 query articles⁶ from the prefiltered set of articles. We left out the query articles used in the training sets of SPECTER and SCINCL.⁷ Next, we sampled candidate pools of various sizes: 200k, 500k, 1M, and 2M. Each candidate pool contained all the articles cited in the query articles, while the remaining candidates were randomly sampled from the prefiltered set. To make the setup more realistic, we considered the publication years of both query and candidate articles: queries were sampled from the articles published in 2019, while candidate articles’ year of publication was 2019 or earlier. Year-based sampling ensures that no article published after the query article can be recommended for citing in that article. Although such year-based sampling still allows for the articles published after the citing (later in 2019) to be included as candidates, it reduces such possibility compared to other benchmarks (e.g., SCIDOCs) that do not account for it.⁸ For each candidate pool size, we repeat the pool sampling procedure three times and report the mean values of the metrics.

Dataset-level results are given in Table 1. We retrieved the top 500 ranked candidates for each model and reported MAP, NDCG, and recall at 30 averaged over three runs for each pool size. For ASPIRE, we used its “BioMed” variant, i.e., the one trained on articles from the biomedicine field.⁹ We optimized BM25’s parameters b and k_1 on separate validation sets constructed in the same way as test sets. A detailed description of the BM25 formula and the role of the two parameters is given in the Appendix A.

We observe that the best performing model on the pool sizes of 200k and 500k is SCINCL, with

⁶In field-level setup, we sampled 200 queries for each of the 19 MAG fields. To keep the total number of queries the same over both setups, we sampled 3800 total queries in the dataset-level setup as well.

⁷At the time of writing, ASPIRE’s training set was not publicly available, so we did not account for that overlap.

⁸Since S2ORC only provides publication years (and not the dates) for articles it contains, filtering can at most be year-based. Additionally, excluding from the candidate pools articles that were published in the same year as the citing would considerably reduce the pool size in some fields.

⁹The other available ASPIRE model, trained on computer science articles, obtained worse results.

Pool sizes →	200k			500k			1M			2M		
	MAP	NDCG	R@30	MAP	NDCG	R@30	MAP	NDCG	R@30	MAP	NDCG	R@30
Models ↓												
BM25	40.4	73.8	43.0	32.8	68.9	36.4	27.4	64.9	31.6	22.5	60.8	26.9
SCIBERT	5.5	40.3	5.6	4.6	38.5	3.9	4.1	37.3	3.0	3.8	36.5	2.2
SPECTER	37.4	72.0	40.9	29.5	66.5	33.9	24.1	62.1	28.7	19.2	57.8	23.8
SCINCL	42.5	75.2	45.2	33.4	69.3	37.6	27.1	64.6	31.9	21.6	60.0	26.5
ASPIRE-BM	41.4	74.7	43.3	32.6	68.9	35.9	25.7	63.5	30.5	20.4	59.0	25.2

Table 1: Results on different pool sizes in the dataset-level setup for BM25 and three considered TAEs. Values in **bold** indicate the best-performing model for a combination of pool size and metric.

ASPIRE and BM25 not far behind. However, with larger pool sizes of 1M and 2M, BM25 performs better than TAEs for most metrics (except R@30 in the 1M pool, where SCINCL outperforms BM25). Given the slight difference in performance between BM25 and SCINCL, our results demonstrate that traditional lexical retrieval is still very competitive in large-scale retrieval scenarios. These results are in line with those of (Reimers and Gurevych, 2021), who also compared the performance of sparse and dense retrieval models on varying pool sizes and found that the performance of the dense retrieval models decreases quicker for the increasing pool sizes compared to sparse methods. Looking at differences between TAEs, the results show clear benefits of finetuning TAEs with inter-article training signal – both SPECTER and SCINCL outperform SCIBERT.

We also observed a significant drop in performance for all the evaluated TAEs compared to their performance on the “Cite” task in SCIDOCs (results on SCIDOCs are given in Appendix A). For example, MAP for SCIBERT in the “Cite” task of SCIDOCs was 48.3 (Cohan et al., 2020), while in a large-scale setup, it ranges from 5.5 in the case of 200k pool size to 3.8 with a 2M pool size. This difference supports our hypothesis that small-scale evaluation is not indicative of the performance of a model in a realistic, large-scale setup. However, our large-scale evaluation results are consistent with some other findings from the evaluation on SCIDOCs, as reported in (Ostendorff et al., 2022): SCINCL’s careful sampling of negatives for the training set leads to a clear improvement in retrieval performance, with SCINCL outperforming SPECTER for all candidate pool sizes.

3.2 Field-level

In the field-level evaluation, we evaluate TAEs on a set of queries and candidate pools from specific scientific fields. Such an evaluation setup resem-

bles a more realistic and also more challenging large-scale retrieval scenario: in a real-world application, given a query article as input, a retrieval model is expected to detect the query article’s field and narrow the candidate pool to articles from that field.

To determine the article’s field, we used Microsoft Academic Graph (MAG) labels provided in S2ORC. We sampled 200 query articles for each of the 19 distinct MAG fields from S2ORC. As in the dataset-level setup, we used year-based splits and sample query articles published in 2019. Next, for each scientific field, we constructed a candidate pool of size 100k that contains all the articles cited in the query articles alongside field-specific negative candidates. To obtain field-specific negative candidates, we randomly sampled the remaining pool articles (up to 100k) from a set of field-cited articles, i.e., a set of articles cited in all S2ORC articles labeled with a specific MAG field. For example, when sampling negative candidates for the Medicine field, we first filtered all articles labeled with Medicine in their S2ORC’s MAG field. We then went through all the articles that they cite and included those in the newly created set of field-cited articles, from which we then sampled negative candidates. As in dataset-level evaluation, we repeat the candidate sampling procedure three times for all the fields where the field-cited article set is larger than 100k (all except Art, History, and Philosophy) and report the mean values of the metrics.

Field-level results in terms of MAP are shown in Table 2 (the NDCG and R@30 results are included in Appendix A; the best performing models are the same in all cases except R@30 for the Bio field). As with dataset-level evaluation, we retrieve the top 500 candidates and report results on these sets. In this setup, we also include ASPIRE-CS, i.e., ASPIRE variant trained on computer science articles.

BM25 achieves the highest mean MAP across

	Art	Bio	Bus	Ch	CS	Eco	Eng	ES	Geog	Geol	His	MS	Mat	Med	Phi	Phy	PS	Psy	Soc	AVG
BM25	36.1	44.3	21.2	42.6	35.6	26.4	34.2	27.5	29.2	32.1	32.8	34.9	35.6	46.3	25.0	35.8	21.2	31.9	17.7	32.1
SCIBERT	6.2	6.7	3.2	6.7	4.2	4.7	4.6	4.8	4.8	4.7	5.3	4.4	4.9	5.4	3.5	5.5	2.8	5.0	3.0	4.8
SPECTER	25.1	37.0	18.0	35.4	33.7	22.6	28.5	22.6	20.1	20.3	17.4	29.0	31.5	48.6	16.2	27.5	14.4	32.0	14.3	26.0
SCINCL	26.9	42.2	18.7	39.6	37.8	23.4	31.5	25.5	23.7	22.7	20.7	30.9	33.5	52.4	18.7	31.4	16.5	34.1	15.7	28.7
ASPIRE-BM	26.8	44.7	19.3	39.9	35.3	24.3	29.7	24.5	23.3	22.9	20.1	29.8	33.1	52.1	17.2	30.0	15.9	33.1	14.3	28.2
ASPIRE-CS	25.8	37.1	20.0	34.9	35.8	23.5	30.2	21.9	21.7	20.1	18.4	27.5	34.2	46.5	17.1	29.0	15.3	32.7	15.7	26.7

Table 2: Results in terms of MAP in the “field-level” evaluation setup. Values in **bold** indicate the best performing model per field. Table with field-abbreviation mapping is given in Table 5 in Appendix A.

all fields, again demonstrating the robust performance of lexical retrieval. SCINCL performs close to BM25, performing best on CS, Med, and Psy fields. While SCINCL’s strong performance in CS and Med fields could be explained by a high percentage of SCINCL training queries from those fields ($\sim 16\%$ and $\sim 25.3\%$ of SCINCL train queries come from Med and CS fields, respectively), a high MAP value in Psy field is unexpected given the small percentage of Psy queries in SCINCL’s train set ($\sim 4.1\%$). Analyzing performance across fields, models perform quite well in some fields (e.g., Med and Bio) and worse in others (e.g., Soc, Bus, PS). Regarding these differences, we note that training sets of most of the TAEs (SPECTER, SCINCL, ASPIRE-BM) have a highly skewed distribution toward Med, Bio, and CS fields. However, another possible explanation might be the different levels of interdisciplinarity in particular fields, which could lead to a richer vocabulary than in mono-disciplinary fields. We leave the investigation of the performance across fields for future work.

Comparing TAEs between each other supports our dataset-level results: SCINCL performs slightly better than ASPIRE (BM), but both outperform SPECTER, which in turn surpasses SCIBERT. Just as in dataset-level evaluation, this ordering is expected given the differences in the training objectives and the training signal used. When comparing different TAEs across fields, we observe that ASPIRE performs especially well in the fields on which it was originally trained: ASPIRE-BM outperformed other TAEs in Bio and Ch fields, which shows that field-specific sentence-level encoders might be more successful than other TAEs for other fields as well. Field-level evaluation results also confirm the need for large-scale evaluation of TAEs – their performance is again much worse than in small-scale benchmark evaluation scenarios, such as SCIDOCs.

To sum up, both of our setups demonstrated (1) a strong performance of a lexical retrieval model BM25, which either surpassed (field-level) or performed competitively to TAEs (dataset-level) in large-scale evaluation scenarios, and (2) a large decrease in performance of all the evaluated TAEs compared to previous small-scale benchmark setups (SCIDOCs). Although we argue that large-scale evaluation is mandatory for more realistic performance estimates, we also recognize the benefits of standardized evaluation benchmarks as they enable the research community to track the improvement on a task easily. However, even when evaluation is not performed on a large scale, we argue that to keep the benchmark-obtained performance estimation as realistic as possible, small benchmarks should contain realistic candidate pools with challenging negatives. With this in mind, in the next section, we describe the construction of a small but more realistic benchmark for evaluating article representations.

4 Multi-Domain Citation Recommendation Benchmark

We now present our newly constructed **Multi-Domain Citation Recommendation** benchmark – **MDCR**. As queries in MDCR, we use the same 200 queries per field as in the field-level evaluation setup (§3.2). For the candidate pools, we start with a random sampling of 5 articles cited in the query article and then select negative candidates.

4.1 Benchmark Construction

To construct challenging candidate pools, we used four different candidate selection strategies: (1) model-based, (2) graph neighbors-based, (3) citation count-based, and (4) random selection. Each candidate strategy produces different candidate types that can be used for a more detailed evaluation of the model’s performance. We outline the selection strategies below.

Model-based selection. This strategy aims to capture difficult candidates for the models evaluated in the large-scale setup. As these candidates are difficult for current models, we expect at least some of these candidates to be challenging for some of the future models. Brown et al. (2019) used a similar method for candidate pool construction in RELISH, where candidates were selected using three different retrieval models and then annotated for relevance by the field experts. In MDCR, we do not provide expert-level annotations for candidates but instead, rely on citations as proxy signals for relevance.

We started with compiling lists of the top 200 candidates per query obtained with each model on the candidate pool from the field-level setup. As we evaluate models that are both trained and used differently, it is reasonable to expect each model to have difficulties with different negative candidates. With this in mind, we intended to select those candidates that are difficult for different models. To determine the degree to which the models’ top candidates overlap, we calculated the average Jaccard index between the highest-ranked negative candidates of different model pairs. The models that obtained a low average Jaccard index tend to make different mistakes (i.e., rank different negative candidates highly) than other models. We chose the three models with the lowest average Jaccard index for selecting negative candidates in this strategy: BM25, SCINCL, and SPECTER. For each query and each of the three selected models, we randomly sampled ten negative candidates from the top 200 highest ranked candidates by the model and added them to the query’s candidate pool. We call these candidate types BM25, SPECTER, and SCINCL for candidates obtained from the respective models.

Graph neighbors-based selection. Research on citation-seeking behavior states that scientists often traverse citation graphs to find articles relevant to their needs (Belter, 2016; Hinde and Spackman, 2015). This suggests that challenging articles should be sampled from the same source, i.e., from a set of articles that either cite or are cited in the articles relevant to the query.

To include such candidates in our pools, we employed the following procedure over the citation graph. Let q be a query article and $OC_q = \{c_1, \dots, c_n\}$ a set of articles that are cited in q (i.e., *outgoing* citations). For each $c_i \in OC_q$,

we constructed corresponding OC_{c_i} and $IC_{c_i} = \{i_1, \dots, i_m\}$ sets, where i_j represents an article that cites c_i (i.e., *incoming* citations). Using such sets, we calculated the overlap similarity as $O_{q,c_i} = |OC_q \cap (OC_{c_i} \cup IC_{c_i})| / |OC_q|$, which represents the similarity between q ’s outgoing citations and c_i ’s incoming and outgoing citations. The high O_{q,c_i} value suggests a considerable overlap in citation links between q and c_i , which indicates that these articles are highly topically related.

We calculated O_{q,c_i} for all the query articles and their cited articles. We then sorted the cited articles by their O_{q,c_i} values, starting from the highest (highly topically relevant) to the lowest (slightly topically related). Since we wanted to make our candidate pool challenging, we started with the c_i that has the highest O_{q,c_i} value and added to the query’s candidate pool all the articles from its $OC_{c_i} \cup IC_{c_i}$ set that are not in OC_q (i.e., cited in the query article). We repeated this procedure until ten negative candidates were added to the pool. We call this candidate type Graph.

Citation count-based selection. In this selection strategy, we created a list of the top 200 most cited articles in each scientific field. We used S2ORC’s MAG field to detect articles from each field and sorted them by the citation counts in descending order. We then randomly sampled ten candidate articles for each query article based on the query article’s MAG field and added these articles to the candidate pool. This type of candidates is called Most cited.

Random selection. Finally, as a less challenging and baseline candidate set, we settled for a random selection strategy, where we randomly sampled ten candidates from the prefiltered set of S2ORC articles. We call this candidate type Random.

4.2 Benchmark Size

Overall, MDCR contains 200 queries per each of the 19 MAG fields, where each query is paired with a set of 60 negative candidates and five cited articles, totalling 247,000 query-candidate pairs that need to be evaluated. Compared to SCIDOCs, where 1,000 queries are paired with candidate pools of size 25 (a total of 25,000 query-candidate pairs), MDCR is almost ten times bigger. While this growth in size increases the computational complexity when using MDCR compared to other smaller benchmarks, it arguably makes the results more realistic. In addition, we also note that since

MDCR is split across different scientific fields, models can be evaluated on specific fields only, which reduces the number of query-candidate pairs to be evaluated.

4.3 Results

Results of evaluation on MDCR are given in Table 3. We report MAP and R@5 (each query is coupled with five positive candidates) across all pairs of the scientific fields and evaluated the model. We evaluate the same set of models as in the field-level large-scale evaluation.

Results demonstrate, yet again, a strong performance from BM25, which outperformed all other models in terms of average metric scores across all fields. Interestingly, when evaluated on MDCR’s small-sized pools, the difference in performance between SCIBERT and other TAEs (e.g., SCINCL) is smaller than in large-scale evaluation (11.7 in MAP on MDCR vs. 37 in MAP on dataset-level, 200k pool size). Such a difference in results confirms the benefits of evaluating TAEs on larger pools to obtain more realistic results. Another observation is a similar average performance between SCINCL and ASPIRE, despite ASPIRE variants being trained only on the articles from specific fields (biomedicine and computer science). As in the field-level evaluation, competitive results from ASPIRE indicate that sentence-level representations might be able to capture a more informative signal between related articles than document-level ones.

Although BM25 outperforms other models in most fields, TAEs obtain the best scores in some cases when looking at performance in specific fields. Specifically, ASPIRE-BM is the top-performing model in the Bio, Med, and Soc fields (and Psy in MAP value), which is not surprising as it was trained on articles from the biomedicine field. Similar goes for ASPIRE-CS and its performance in the Mat field, although it does not yield the best results in the field it was trained on (CS). However, when analyzing ASPIRE’s performance, it is worth noting that we did not account for the overlap of ASPIRE training queries with our new benchmark since ASPIRE’s training set was not publicly available at the time of writing. For this reason, the results of both ASPIRE variants might be too optimistic if the train-test overlap is significant.

4.4 Performance across Candidate Types

To analyze the difficulty of candidate types that we introduced in §4.1, we evaluate the models on

subsets of candidate pools consisting of 5 cited articles and all negative candidates from specific candidate type. Evaluation across candidate types allows us to analyze how difficult each candidate type is for each model. As the candidates obtained via model-based selection are chosen precisely because they were difficult for the particular models, we do expect these models to not perform well on such candidates. However, such evaluation can reveal interesting insights into the differences across the evaluated models, e.g., whether the same candidate types are difficult for all neural-based models.

Results of this evaluation are presented in Table 4. Unsurprisingly, Random candidates are the easiest candidate type for all the evaluated models. Candidates from the Most cited type are also relatively easy for the models, with on average >90 score in MAP. On average, the most challenging candidate type is the Graph candidates subset, with an average MAP score of 55.9. Interestingly, the best-performing model on the Graph candidates subset is SCINCL, which explicitly uses citation graph embeddings in selecting training examples. Such a training strategy seems to help the model distinguish between relevant and irrelevant graph neighbors.

The performance on the candidate types obtained with the model-based selection strategy differs between TAEs and BM25, which is somewhat expected given the difference between neural (TAEs) and non-neural (BM25) models. As expected, negative candidates from the BM25 type are the most difficult for BM25 itself since those were sampled from a set of top candidates provided by BM25. On the other hand, TAEs (SCIBERT excluded) all perform similarly well on the BM25 candidate type. Likewise, SPECTER and SCINCL candidates are the most difficult for SPECTER and SCINCL, respectively, while BM25 performs better than TAEs on these candidate types. It is interesting to note the difference in the performance of SCINCL on SPECTER candidates compared to the performance of SPECTER on SCINCL candidates. While SCINCL outperforms SPECTER on SPECTER candidates with more the 10 points in the absolute value (for both metrics), SPECTER improves over SCINCL on SCINCL candidates with only 3.7 absolute points. These results again confirm that the way in which negative candidates are sampled when training the models with the contrastive learning objective is important. As for

Models →	BM25		SciBERT		SPECTER		SciNCL		ASPIRE-BM		ASPIRE-CS	
Fields ↓	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5
Art	38.2	32.3	22.4	16.6	34.1	28.8	34.7	29.2	34.0	27.7	34.1	28.0
Bio	38.3	33.6	20.4	14.0	34.6	30.0	36.8	32.3	38.7	33.7	35.7	29.9
Bus	28.1	22.5	19.1	13.1	27.5	21.8	28.5	24.6	28.5	23.4	29.6	23.1
Ch	38.0	32.6	20.0	13.7	33.7	29.3	36.5	31.5	36.5	31.0	34.1	28.3
CS	34.8	30.5	19.5	12.7	35.6	30.4	37.2	32.2	35.4	30.4	35.4	30.1
Eco	30.5	26.0	21.4	15.4	27.3	21.9	28.3	23.2	29.3	24.3	28.0	22.7
Eng	34.6	29.3	20.5	13.9	31.3	27.3	34.2	28.0	32.7	27.7	33.4	28.1
ES	31.6	26.2	21.3	15.1	30.1	24.2	31.5	25.5	30.8	24.7	29.9	23.7
Geog	31.8	27.8	21.9	16.7	26.4	22.2	29.5	23.8	30.3	26.0	28.4	22.2
Geol	33.1	28.0	19.5	13.9	24.8	20.1	25.7	19.9	28.5	23.5	25.8	21.4
His	38.1	32.9	20.8	15.2	27.1	20.6	30.9	23.9	31.0	24.2	28.5	22.1
MS	36.1	30.7	22.1	15.5	34.1	28.2	35.8	29.6	35.8	29.8	34.0	29.2
Mat	35.3	28.3	22.8	18.3	34.2	28.9	34.9	30.1	36.2	31.0	36.9	32.2
Med	38.6	32.5	22.0	16.4	41.4	36.3	42.7	36.5	44.0	37.8	41.7	36.7
Phi	30.2	25.7	19.2	13.3	27.1	21.1	29.9	23.5	28.7	24.1	29.1	23.3
Phy	35.1	30.2	23.9	18.1	30.8	26.3	34.5	30.3	32.9	27.7	32.9	28.7
PS	28.6	23.1	19.4	14.0	24.2	18.0	26.4	21.7	25.9	21.2	26.8	21.7
Psy	32.5	28.9	20.3	16.2	32.3	28.1	34.2	30.5	34.3	29.4	34.2	28.3
Soc	26.8	20.5	20.2	15.8	25.2	20.5	26.7	21.9	27.3	22.2	26.7	22.2
AVG	33.7	28.5	20.9	15.2	30.6	25.5	32.6	27.3	32.7	27.4	31.8	26.4

Table 3: Results in terms of MAP and R@5 on MDCR. Values in **bold** indicate the best performing model for a combination of field and metric.

Candidate types →	BM25		SPECTER		SciNCL		Graph		Most cited		Random	
Models ↓	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5	MAP	R@5
BM25	52.2	39.0	68.8	57.5	68.0	56.9	58.0	46.7	90.3	82.2	93.0	86.3
SciBERT	50.5	39.1	47.6	36.0	49.0	37.7	47.2	36.1	79.6	69.3	84.9	75.8
SPECTER	67.4	56.3	51.1	38.7	57.8	45.8	57.3	46.3	92.8	86.5	99.0	97.1
SciNCL	68.3	57.6	61.2	49.8	54.1	42.1	58.1	47.3	94.0	88.3	99.0	97.2
ASPIRE-BM	66.7	55.6	57.7	46.5	59.3	47.5	57.3	46.5	93.6	87.6	99.1	97.2
ASPIRE-CS	66.0	54.9	55.7	43.8	58.5	46.8	57.2	46.6	94.3	88.7	98.9	96.8
AVG	61.8	50.4	57.0	45.4	57.8	46.1	55.9	44.9	90.8	83.8	95.7	91.7

Table 4: Results in terms of MAP and R@5 for different candidate types on MDCR.

the ASPIRE variants and candidate types obtained with the model-based strategy, ASPIRE variants perform better on the BM25 candidate type than on the SPECTER or SciNCL type. We hypothesize that such difference is due to TAEs being similar neural models and therefore prone to similar errors regarding semantic vs. lexical matching of texts. In contrast, BM25, a purely lexical model, makes different errors. We leave the analysis of the differences in performance between neural and non-neural models for future work.

5 Conclusion

We evaluated transformer-based article encoders in large-scale citation recommendation scenarios across different scientific fields and candidate pool sizes. Together with transformer-based encoders, we evaluated the performance of a robust lexical retrieval baseline BM25 and demonstrated that it still

performs competitively with recent neural-based models. In the case of large field-specific candidate pools, BM25 outperformed transformer-based models in most fields.

Furthermore, to promote a more realistic and a more diverse evaluation across different fields in comparison to the existing benchmarks used for evaluating scientific article representations, we presented a new multi-domain benchmark dataset based on citation recommendation task, which we call MDCR. Evaluation on MDCR demonstrated the difficulty of specific candidate types and set the ground for evaluating future scientific article encoders.

Our evaluation demonstrated the varying performance across scientific fields, which we believe should be analyzed in future work to improve encoders’ performance across all fields, not just those prevailing in the datasets. Given that our bench-

mark dataset is not expert-annotated but rather based on citations as relevance signals, we propose constructing an expert-annotated dataset with articles from different scientific fields. We hope our contributions will stimulate the community to work on more realistic and challenging evaluation setups of scientific article recommendation models.

Acknowledgments

This research was supported by the grant AIDWAS KK.01.2.1.02.0285. The first author has been supported by a grant from the Croatian Science Foundation (HRZZ-DOK-2018-09).

References

- Zafar Ali, Guilin Qi, Khan Muhammad, Pavlos Kefalas, and Shah Khusro. 2021. Global Citation Recommendation Employing Generative Adversarial Network. *Expert Systems with Applications*, 180:114888.
- 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Christopher W Belter. 2016. Citation Analysis as a Literature Search Method for Systematic Reviews. *Journal of the Association for Information Science and Technology*, 67(11):2766–2777.
- Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. **Content-Based Citation Recommendation**. 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 238–251, New Orleans, Louisiana. Association for Computational Linguistics.
- Peter Brown, Aik-Choon Tan, Mohamed A El-Esawi, Thomas Liehr, Oliver Blanck, Douglas P Gladue, Gabriel MF Almeida, Tomislav Cernava, Carlos O Sorzano, Andy WK Yeung, et al. 2019. Large Expert-curated Database for Benchmarking Document Similarity Detection in Biomedical Literature Search. *Database*, 2019.
- Xiaoyan Cai, Sen Liu, Libin Yang, Yan Lu, Jintao Zhao, Dinggang Shen, and Tianming Liu. 2022. COVIDSum: A Linguistically Enriched SciBERT-based Summarization Model for COVID-19 Scientific Papers. *Journal of Biomedical Informatics*, 127:103999.
- Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. **SPECTER: Document-level Representation Learning using Citation-informed Transformers**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nianlong Gu, Yingqiang Gao, and Richard H. R. Hahnloser. 2022. Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-Based Reranking. In *Advances in Information Retrieval*, pages 274–288, Cham. Springer International Publishing.
- Tanishq Gupta, Mohd Zaki, NM Krishnan, et al. 2022. MatSciBERT: A Materials Domain Language Model for Text Mining and Information Extraction. *npj Computational Materials*, 8(1):1–11.
- Sebastian Hinde and Eldon Spackman. 2015. Bidirectional Citation Searching to Completion: an Exploration of Literature Searching Methods. *Pharmacoeconomics*, 33(1):5–11.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics*, 36(4):1234–1240.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.
- Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. 2015. Verboseness Fission for BM25 Document Length Normalization. In *Proceedings of the 2015 International Conference on the Theory of Information Retrieval*, pages 385–388.

- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. [S2ORC: The Semantic Scholar Open Research Corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.
- 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, Online. Association for Computational Linguistics.
- Yuanhua Lv and ChengXiang Zhai. 2011. Adaptive Term Frequency Normalization for BM25. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1985–1988.
- Zoran Medić and Jan Šnajder. 2020. A Survey of Citation Recommendation Tasks and Methods. *Journal of computing and information technology*, 28(3):183–205.
- Sheshera Mysore, Arman Cohan, and Tom Hope. 2021a. [Multi-Vector Models with Textual Guidance for Fine-Grained Scientific Document Similarity](#).
- Sheshera Mysore, Tim O’Gorman, Andrew McCallum, and Hamed Zamani. 2021b. [CSFCube - A Test Collection of Computer Science Research Articles for Faceted Query by Example](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Usman Naseem, Katarzyna Musial, Peter Eklund, and Mukesh Prasad. 2020. Biomedical Named-entity Recognition by Hierarchically Fusing BioBERT Representations and Deep Contextual-level Word-embedding. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Rodrigo Nogueira, Zhiying Jiang, Kyunghyun Cho, and Jimmy Lin. 2020. Evaluating Pretrained Transformer Models for Citation Recommendation. In *CEUR Workshop Proceedings*, volume 2591, pages 89–100. CEUR-WS.
- Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. *arXiv preprint arXiv:2202.06671*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2021. [The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 605–611, Online. Association for Computational Linguistics.
- Stephen E Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-poisson Model for Probabilistic Weighted Retrieval. In *SIGIR’94*, pages 232–241. Springer.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ellen Voorhees, Tasmee Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. [TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection](#). *SIGIR Forum*, 54(1).
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or Fiction: Verifying Scientific Claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

A Appendix

A.1 BM25

Here we provide details about the relevance score function used by BM25 (Robertson and Walker, 1994). Before calculating relevance scores for pairs of articles, article texts are first transformed into bag-of-words vectors. Given a query Q , containing terms q_1, \dots, q_n , and a document D , BM25 calculates relevance score s as follows:

$$s(Q, D) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

where $f(q_i, D)$ is the frequency q_i in document D , $|D|$ is the length of D in words, avgdl is the average document length, and k_1 and b are parameters that can be tuned for a specific document collection. $\text{IDF}(q_i)$ is the inverse document frequency for q_i , and is typically calculated as:

$$\text{IDF}(q_i) = \ln \left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right)$$

where N is the total number of documents in the collection and $n(q_i)$ is the number of documents containing the term q_i .

Intuitively, s will output higher scores for a document D that contains many terms as Q , that also do not appear often in other documents. When it comes to parameters b and k_1 , b controls to what degree the length of a document will affect the final score (Lipani et al., 2015), while k_1 controls to what degree an additional occurrence of a term affects the final score (Lv and Zhai, 2011).

A.2 Field Abbreviations

Table 5 shows the abbreviations for MAG fields.

A.3 Evaluation on SCIDOCs

Table 6 shows the results of evaluation of SCIBERT, SPECTER, SCINCL, and ASPIRE on SCIDOCs benchmark, as reported in the previous work.

A.4 Field-level Evaluation Results

Results for “field-level” evaluation setup in terms of NDCG and recall@30 are given in Tables 7 and 8, respectively. Best scoring combinations of model and field are mostly the same as in case of MAP (reported in Table 2), with the exception of

MAG field	Abbreviation
Art	Art
Biology	Bio
Business	Bus
Chemistry	Ch
Computer Science	CS
Economics	Eco
Engineering	Eng
Environmental Science	ES
Geography	Geog
Geology	Geol
History	His
Materials Science	MS
Mathematics	Mat
Medicine	Med
Philosophy	Phi
Physics	Phy
Political Science	PS
Psychology	Psy
Sociology	Soc

Table 5: Abbreviations for MAG fields that we use in the field-level evaluation and in the new benchmark.

Model	MAP	NDCG
SCIBERT	48.3	71.7
SPECTER	88.3	94.9
SCINCL	93.6	97.3
TS-ASPIRE	91.0	95.0

Table 6: Results of different TAEs evaluated on SCIDOCs’s “Cite” task. Values as reported in (Cohan et al., 2020), (Mysore et al., 2021a), and (Ostendorff et al., 2022).

recall@30 in Bio field, where BM25 yields the best result (as opposed to ASPIRE-BM in case of MAP).

Model	Art	Bio	Bus	Ch	CS	Eco	Eng	ES	Geog	Geol	His	MS	Mat	Med	Phi	Phy	PS	Psy	Soc	AVG
BM25	64.2	78.8	56.9	76.2	69.3	65.0	68.0	62.6	64.5	69.2	62.7	69.8	70.1	77.8	56.9	70.1	56.2	70.2	54.0	66.5
SCIBERT	33.2	45.8	33.6	43.7	36.0	39.6	36.4	36.8	37.6	39.6	32.9	37.3	37.9	39.8	31.2	39.5	32.1	40.9	32.9	37.2
SPECTER	53.5	74.2	53.9	71.8	67.7	61.9	63.6	58.2	56.5	59.2	47.0	65.6	66.5	79.2	48.6	63.8	49.2	70.5	49.9	61.1
SCINCL	55.3	77.5	54.1	74.0	70.3	62.5	66.0	60.7	59.6	61.2	50.5	67.0	68.0	81.1	50.8	67.1	51.4	71.7	51.3	63.1
ASPIRE-BM	55.6	79.2	55.2	74.4	69.1	63.5	64.8	60.0	59.4	61.5	50.1	66.2	68.1	81.0	50.1	66.2	50.6	70.9	49.8	62.9
ASPIRE-CS	54.8	74.2	56.1	71.1	69.4	62.9	65.0	57.7	57.8	59.0	48.1	64.3	69.1	77.9	49.6	65.5	50.4	70.8	51.4	61.8

Table 7: Results in terms of NDCG in the “field-level” evaluation setup. Values in **bold** indicate the best performing model per field.

Model	Art	Bio	Bus	Ch	CS	Eco	Eng	ES	Geog	Geol	His	MS	Mat	Med	Phi	Phy	PS	Psy	Soc	AVG
BM25	46.5	38.9	27.5	41.7	43.1	29.7	42.7	35.3	34.6	34.2	42.7	39.5	42.2	48.2	34.0	40.9	28.3	32.0	24.0	37.2
SCIBERT	9.0	4.8	2.1	5.8	5.0	3.3	5.5	5.1	5.0	3.9	8.0	4.1	6.0	4.8	3.8	6.6	1.6	3.4	2.4	4.8
SPECTER	35.5	33.4	24.7	35.3	41.9	25.7	36.4	30.3	24.3	23.4	24.8	34.2	38.5	49.0	24.1	33.4	19.9	32.6	19.8	30.9
SCINCL	38.9	36.8	25.4	39.3	46.4	26.0	39.5	33.0	28.0	25.5	28.2	35.5	40.2	52.0	26.4	37.1	23.0	34.7	21.5	33.5
ASPIRE-BM	34.9	38.2	25.1	38.6	42.5	27.3	37.0	31.5	27.5	25.3	28.0	34.5	39.9	51.6	25.0	35.2	22.2	32.3	19.2	32.4
ASPIRE-CS	34.9	33.0	25.5	34.7	42.1	25.9	37.5	29.4	25.6	23.2	25.3	33.0	40.1	47.3	24.7	34.5	21.4	32.2	20.1	31.1

Table 8: Results in terms of recall@30 in the “field-level” evaluation setup. Values in **bold** indicate the best performing model per field.