# Evaluating Pretrained Transformer Models for Citation Recommendation

Rodrigo Nogueira,[1,2] Zhiying Jiang,[2] Kyunghyun Cho,[3,4,5,6] Jimmy Lin[2]

[1] Tandon School of Engineering, New York University
[2] David R. Cheriton School of Computer Science, University of Waterloo
[3] Courant Institute of Mathematical Sciences, New York University
[4] Center for Data Science, New York University
[5] Facebook AI Research
[6] CIFAR Azrieli Global Scholar

**Abstract.** Citation recommendation systems for the scientific literature, to help authors find papers that should be cited, have the potential to speed up discoveries and uncover new routes for scientific exploration. We treat this task as a ranking problem, which we tackle with a two-stage approach: candidate generation followed by re-ranking. Within this framework, we adapt to the scientific domain a proven combination based on "bag of words" retrieval followed by re-scoring with a BERT model. We experimentally show the effects of domain adaptation, both in terms of pretraining on in-domain data and exploiting in-domain vocabulary. In addition, we evaluate eleven pretrained transformer models and analyze some unexpected failure cases. On three different collections from different scientific disciplines, our models perform close to or at the state of the art in the citation recommendation task.

## 1 Introduction

The volume of scientific publications is growing at an incredible rate. For example, over 900,000 papers are added per year to MEDLINE, a database of the life sciences and biomedical literature.[1] A recent study estimates that 3M papers are published annually in the English language, with a growth rate of 3–5% per year [18]. This flood of information has made it nearly impossible for researchers to keep abreast of discoveries and innovations, both in their specific sub-field as well as more broadly. Furthermore, there is an overwhelming amount of material that a scientist entering a new field of study needs to read before becoming familiarized with common concepts, methods, and other foundations.

A number of tools have come along to help researchers cope with this deluge. For example, keyword-based literature search engines (Google Scholar, Microsoft Academic, PubMed, and Semantic Scholar) and citation recommendation



[1] https://www.nlm.nih.gov/bsd/stats/cit_added.html

tools [5, 2, 27, 21, 14] help scientists find relevant articles, often exploiting citation networks to identify what's important in a particular field. Methods to automatically populate scientific knowledge bases [12, 34, 35] form another broad approach to tackling this challenge.

In this work, we investigate the potential of deep pretrained transformer models such as BERT [7] and large scientific datasets such as Open Research [1] to improve scientific search tools. More concretely, we tackle the task of scientific literature recommendation, where a paper (title and abstract) is given as a query, and the system's task is to find papers that should be cited. We use a standard keyword search engine (based on inverted indexes) with BM25 ranking [33] to initially retrieve candidate documents and evaluate various pretrained transformer models as re-rankers.

We find that this simple pipeline is more effective than previous cluster-based methods [32, 4]. To summarize, our main contributions are as follows:

- We evaluate eleven pretrained ranking models and find that pretraining on the target domain and using domain-specific vocabulary leads to large improvements over a general-purpose model.
- We find that despite the effectiveness of the pretrained transformer models as query–document relevance estimators, they perform poorly when the term overlap between the query and candidate documents is low. To address this issue, we train with more query–candidate pairs that have low term overlap, but interestingly, such a model performs poorly, even on the training set (see Section 5.2).
- Contrary to our expectation given the symmetric nature of query and candidate documents, we find that query terms are more important than candidate document terms for relevance estimation (see Section 5.3).

## 2 Related Work

Most early methods for scientific literature search and recommendation take advantage of keyword-based retrieval [13, 22]. These techniques suffer from the term mismatch problem, which is common in "bag-of-words" retrieval methods, but the issue is aggravated by the diversity of scientific vocabulary [17, 8, 29]. As the number of users grows, popular search engines can exploit interaction signals to learn better ranking models [28, 11, 10]. However, the reported gains are relatively small compared to classic ranking methods such as BM25.

Another common approach in scientific recommendation systems is collaborative filtering [27, 24, 6]. These methods typically suffer from the cold-start problem, in which there is not enough evidence about new items (or users) to make predictions accurately.

More recently, cluster-based methods have started to become competitive with traditional retrieval-based methods in this task. Kanakia et al. [19] cluster papers based on their word embedding representation and use co-citations to alleviate the cold-start problem. However, they perform human evaluations on a private dataset, which excludes an empirical comparison to our approach.

Perhaps closest to our work is Eto [9], who uses a combination of proximity measures from the graph of co-citations to score candidate documents. The edges in the graph are weighted by the distance in which two citations occur in the citing document. This method requires access to the full text of the citing document, which is often not available (for example, due to paywalled content). Our method, on the other hand, predicts citations using only article abstracts, which are widely available in scientific corpora.

The methods described so far and our work fall in the category of *global* methods, which aim at recommending citations for the entire paper. Another category comprises *local* methods, which aim at recommending citations for a specific sentence or paragraph in the document [14, 26, 15, 16]. We do not compare our method to these as we do not assume access to the full text.

## 3 Methods

This work tackles the task of citation recommendation: given a partially written paper, the system's task is to return all papers that should be cited in it. The input query $q$ is the title and abstract of a paper (and *not* the full text). We argue that this assumption is crucial to building a useful tool as authors might desire recommendations of relevant citations prior to writing most of their paper.

Our method comprises two phases, *Retrieval* and *Ranking*. In the first phase, the top-$k$ papers $D$ are retrieved by a keyword search engine when queried with query $q$. In the second phase, we compute the probability $p(d|q)$ of each paper $d \in D$ being relevant to $q$. For this, we use a BERT [7] re-ranker model based on Nogueira and Cho [30]. Using the same notation as Devlin et al., we feed the query tokens as sequence A and the candidate paper tokens as sequence B.

In our setup, both the query and the candidate are the concatenation of the title and abstract of each paper, resulting in an input sequence that is often longer than the maximum tokens allowed by the model (typically 512 tokens). To handle this, we devote 256 tokens for the query and 256 for the candidate, truncating as necessary. At inference time, we use the model as a binary classifier: we feed the [CLS] token to a single layer neural network to obtain $p(d|q)$. The output of our method is a list of papers $D$ ranked by $p(d|q)$. Training details are provided in Section 4.2.

## 4 Experimental Setup

### 4.1 Datasets

**Open Research.** We train and evaluate our models on the Open Research corpus [1],[2] comprising 7.2M computer science and biomedical paper abstracts and their references. We closely follow the data processing steps from Bhagavatula

[2] `https://s3-us-west-2.amazonaws.com/ai2-s2-research-public/open-corpus/2017-02-21/papers-2017-02-21.zip`

**Table 1.** Statistics of the datasets.

| | Open Research | DBLP | PubMed |
|---|---|---|---|
| Total # of docs | 6,892,252 | 50,227 | 47,347 |
| Total # of citations | 44,400,729 | 156,807 | 825,371 |
| Avg. # citations per doc | 6.45 | 3.12 | 17.43 |
| Avg. len. per doc (char) | 1,391 | 1,193 | 1,504 |
| Queries - Train | 3,343,809 | 27,322 | 26,793 |
| - Dev | 487,582 | 8,324 | 2,768 |
| - Test | 464,449 | 931 | 8,815 |
| q/rel. doc pairs - Train | 32,470,673 | 106,011 | 558,674 |
| - Dev | 5,985,787 | 38,628 | 66,655 |
| - Test | 5,944,269 | 12,168 | 200,042 |

et al. [4] to create the training, development, and test sets. In more detail, we sort papers by publication year and use the oldest 80% for training (1991–2014), the next 10% for development (2014–2015), and the most recent 10% for testing (2015–2016). Since the development and test sets are too large (400k+ papers), we randomly sample 20k examples from each set. We remove papers that do not cite any other paper or that have no year of publication. Finally, we remove citations of papers that are not in the corpus or whose year of publication is later than that of the citing paper. Table 1 shows the statistics of the final dataset after all processing steps.

Note that although our dataset statistics do not match those reported in Bhagavatula et al. [4], they match the output of the evaluation script provided by the authors.[3] The difference is that the authors report statistics before the filtering steps (e.g., removing papers without references). Thus, our corpus and dataset splits match exactly and thus our results are comparable.

**DBLP and PubMed.** The DBLP and PubMed datasets were introduced by Ren et al. [32] and comprise papers from computer science and biomedicine, respectively. We apply the same data processing steps from Bhagavatula et al., and the resulting dataset statistics are summarized in Table 1.

Once processed in the manner described above, the citations within each paper serve as the ground truth for that paper. That is, using a specific paper as a query, the perfect results set comprises the actual citations in that paper.

When evaluating our method on DBLP and PubMed, we use models trained on Open Research's training set as this yields better results than training on the much smaller DBLP and PubMed training sets. To avoid leaking training data into the evaluation sets, we use the following method to remove documents in

[3] `https://github.com/allenai/citeomatic/blob/master/citeomatic/scripts/evaluate.py`

Open Research's training set that appear in the development and test sets of PubMed and DBLP: We remove special characters from the title and use Jaccard similarity (on unigrams) to calculate the closeness of two documents, filtering with a threshold of 0.7. This method results in approximately half of the papers in the development and test sets of PubMed and DBLP being removed from the training set of Open Research.

### 4.2 Re-ranker Training

To obtain the positive and negative examples used to train our binary classification models, we retrieve the top 10 papers for each query (title + abstract) using the Anserini IR toolkit[4] [36, 37] with BM25 ranking. Among these, approximately 6% on average are relevant papers (positive examples). We do not balance positive and negative examples; see additional discussions about this decision in Section 5.2.

Starting with a pretrained BERT model, we fine-tune it to our task using cross-entropy loss:

$$L = -\sum_{j \in J_{\text{pos}}} \log(p(d_j|q)) - \sum_{j \in J_{\text{neg}}} \log(1 - p(d_j|q)), \tag{1}$$

where $J_{\text{pos}}$ and $J_{\text{neg}}$ are the indexes of the relevant and non-relevant papers and $p(d_j|q)$ is the relevance probability the model assigns to the $j$-th paper. We examine several BERT variants, detailed in Section 5.1.

All models are fine-tuned using Google's TPUs v3-8 with a batch size of 128 (128 sequences $\times$ 512 tokens = 65,536 tokens/batch) for 300k iterations, which takes approximately three days. This corresponds to training on 38.4M (300k $\times$ 128) query–candidate pairs, or 1.1 epochs. We do not see any improvements in the development set when training for another 700k iterations, which is equivalent to 3.8 epochs. We use Adam [20] with the initial learning rate set to $3 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay of 0.01, learning rate warm-up over the first 10,000 steps, and linear decay of the learning rate. We use a dropout probability of 0.1 in all layers.

### 4.3 Inference and Metrics

At inference time, we first retrieve the top 1000 candidate documents with the title and abstract as the query using BM25 ranking in Anserini. These documents are further re-ranked with one of the variants of the fine-tuned BERT models (see Section 5.1 for more details). Following Bhagavatula et al. [4], we evaluate the results using $F_1$ of the top 20 retrieved papers ($F_1$@20) and Mean Reciprocal Ranking (MRR) of the top 1000 retrieved papers. We additionally report Recall@1000 (R@1000) to assess the effectiveness of our keyword search in isolation, which provides an upper bound on re-ranking effectiveness.

[4] `http://anserini.io/`

**Table 2.** Main results on Open Research, DBLP, and PubMed.

| | $F_1$@20 | | MRR | | R@1000 | |
|---|---|---|---|---|---|---|
| | Dev | Test | Dev | Test | Dev | Test |
| **Open Research** | | | | | | |
| BM25 [4] | - | 0.058 | - | 0.218 | - | - |
| BM25 (Anserini) | 0.082 | 0.089 | 0.279 | 0.312 | 0.424 | 0.421 |
| Citeomatic [4] | - | 0.125 | - | 0.330 | - | - |
| BM25 + SciBERT-Large | 0.136 | 0.132 | 0.430 | 0.431 | 0.424 | 0.421 |
| **DBLP** | | | | | | |
| BM25 [4] | - | 0.119 | - | 0.425 | - | - |
| BM25 (Anserini) | 0.105 | 0.194 | 0.352 | 0.585 | 0.669 | 0.691 |
| ClusCite [32] | - | 0.237 | - | 0.548 | - | - |
| Citeomatic [4] | - | **0.303** | - | 0.689 | - | - |
| BM25 + SciBERT-Large | 0.149 | 0.272 | 0.472 | **0.714** | 0.669 | 0.691 |
| **PubMed** | | | | | | |
| BM25 [4] | - | 0.209 | - | 0.574 | - | - |
| BM25 (Anserini) | 0.299 | 0.268 | 0.793 | 0.721 | 0.794 | 0.765 |
| ClusCite [32] | - | 0.274 | - | 0.578 | - | - |
| Citeomatic [4] | - | **0.329** | - | 0.771 | - | - |
| BM25 + SciBERT-Large | 0.326 | 0.304 | 0.835 | **0.792** | 0.794 | 0.765 |

## 5 Results

Our main results are shown in Table 2 with SciBERT-Large as the ranking model, selected based on the experiments in Section 5.1. On the Open Research dataset, our best configuration (BM25 + SciBERT-Large) improves upon the best previous result in terms of both $F_1$@20 and MRR. On the smaller DBLP and PubMed datasets, our method is on par with the state of the art. Note that our BERT-based models are trained only on Open Research as we achieve better results than training on the smaller datasets.

Interestingly, our baseline BM25 implementation using Anserini out of the box, denoted "BM25 (Anserini)" in Table 2, is 3–7 points higher in $F_1$@20 than the BM25 implementation of Bhagavatula et al. This is likely due to the choice of the query form that we use for "bag of words" retrieval, which is analyzed in Section 5.3, and perhaps a better implementation of BM25 in Anserini (which is based on Lucene).

Our method appears to be as effective and more scalable than a cluster-based approach. For example, Bhagavatula et al.'s model requires at least 100

**Table 3.** Results on Open Research's development set of BERT-based models pretrained under different settings. All models are fine-tuned for approximately one epoch on the training set.

| | Pretrained Model | Size | Pretraining Corpus | Tokens | Vocabulary | Cased | $F_1$@20 | MRR |
|---|---|---|---|---|---|---|---|---|
| (1) | NCBI | Base | PubMed+MIMIC | 4.5B | Wiki+Books | | 0.093 | 0.315 |
| (2) | NCBI | Large | PubMed+MIMIC | 4.5B | Wiki+Books | | 0.105 | 0.352 |
| (3) | Google | Base | Wiki+Books | 3.3B | Wiki+Books | | 0.113 | 0.374 |
| (4) | Google | Large | Wiki+Books | 3.3B | Wiki+Books | | 0.115 | 0.373 |
| (5) | Google WWM | Large | Wiki+Books | 3.3B | Wiki+Books | | 0.121 | 0.399 |
| (6) | RoBERTa | Large | Various (Non-Scientific) | 33B | (Non-Scientific) | | 0.125 | 0.409 |
| (7) | BioBERT v1.1 | Base | Wiki+Books+PubMed+PMC | 21.3B | PubMed+PMC | ✓ | 0.128 | 0.417 |
| (8) | SciBERT | Base | Open Research (1M Full Papers) | 3.2B | Wiki+Books | | 0.125 | 0.409 |
| (9) | SciBERT | Base | Open Research (1M Full Papers) | 3.2B | Open Research | | 0.131 | 0.423 |
| (10) | SciBERT | Large | Open Research (7M Abstracts) | 1.4B | Wiki+Book | | 0.135 | 0.420 |
| (11) | SciBERT | Large | Open Research (7M Abstracts) | 1.4B | Open Research | | **0.137** | **0.430** |

GB of RAM to search the 7M documents in the Open Research corpus,[5] whereas keyword search has far more modest memory requirements.

In the next sections, we investigate the effectiveness of our method by evaluating various pretrained transformer models, as well as the effects of class imbalance and different query forms.

### 5.1 In- vs. Out-Domain Pretraining

Here we investigate how different pretraining configurations change effectiveness in the target task. The results, shown in Table 3, are from fine-tuning the pretrained models on Open Research's training set for 300k iterations with a batch size of 128, which corresponds to approximately 1.1 epochs. In the remainder of this paper, we call an *in-domain* corpus a collection whose majority of documents are from the same domains as those in Open Research (i.e., biomedicine and computer science), and we call an *out-domain* corpus a collection whose majority of papers are not from those domains.

The models pretrained on an in-domain corpus, i.e., BioBERT [23] (row 7) and SciBERT [3] (rows 8–11), yield significant improvements in the target task over models pretrained on a corpus of a similar size but a different domain (rows 3–5). Pretraining on an out-domain corpus ten times the size of the in-domain corpus results in lower effectiveness on the target task; compare RoBERTa [25], row 6 vs. row 10. We conclude that, at least for the task of citation recommendation, pretraining on a smaller in-domain corpus is more effective than pretraining on a larger out-domain corpus.

When pretraining settings are kept the same except for the vocabulary, the use of in-domain vocabulary gives 5–10% improvement over out-domain vocab-

[5] `https://github.com/allenai/citeomatic#citeomatic-evaluation`

ulary (row 8 vs. 9 and row 10 vs. 11). This make intuitive sense, and Beltagy et al. [3] report a similar finding in other tasks as well.

The NCBI models [31] (rows 1 and 2) are pretrained on an in-domain corpus but produce worse results than models pretrained on an out-domain corpus of a similar size (rows 3–5). They also underperform when compared to SciBERT-Base (row 8), which is pretrained on an in-domain corpus of a similar size but comprises full papers instead of abstracts. As also noted by Beltagy et al. [3], this result suggests that pretraining with longer documents improves the target task effectiveness.

We find that model size appears to be even more important than document length. Our SciBERT-Large models (rows 10 and 11) have higher effectiveness than the SciBERT-Base models (rows 8 and 9) despite being pretrained on a smaller corpus of 7M paper abstracts (1.4B tokens) as opposed to 1M full-text papers (3.2B tokens).

### 5.2 Class Imbalance

Because we only use the top 10 papers returned by BM25 as training examples, the BERT-based models in this work are trained with more negative examples than positive ones (94% vs. 6%). In a separate experiment, to balance these classes, we include in the training phase pairs of query and relevant papers not retrieved by BM25, but this results in $F_1$@20 and MRR close to zero in both training and development sets. We obtain a similar result when adding to the training set negative candidates randomly sampled from the corpus.

What explains these findings? We hypothesize that although BERT is a strong model for document ranking, it still partly relies on exact term match to learn relevance. Thus, when we sample training documents *not* using an exact term match method such as BM25, fewer terms between the query and the candidate paper match, which makes learning relevance harder. Further studies should investigate if this limitation applies to other tasks as well.

### 5.3 Query Analysis

In the citation recommendation task, the "query" used for initial retrieval can take many forms, such as the title of the paper, the concatenation of title and abstract, or keywords extracted from the text. Here we investigate how these query forms impact the effectiveness of a keyword-based retrieval method.

In Table 4, we show the effectiveness of BM25 on the Open Research development set. For *Key Terms*, we follow Bhagavatula et al. [4] and use Whoosh[6] to first create an index and then extract key terms from the title and abstract with Whoosh's `key_terms_from_text` method. Despite being faster due to having fewer query terms, the results show that this method has lower effectiveness than simply concatenating the title and abstract of the paper.

[6] `https://whoosh.readthedocs.io/en/latest/`

**Table 4.** BM25 results on Open Research's development set when different query forms are used. BERT-based re-ranking is not applied in these experiments.

| | Open Research | | | PubMed | | | DBLP | | |
|---|---|---|---|---|---|---|---|---|---|
| Query Type | $F_1$@20 | MRR | R@1000 | $F_1$@20 | MRR | R@1000 | $F_1$@20 | MRR | R@1000 |
| Key Terms (Whoosh) | 0.065 | 0.251 | 0.282 | 0.201 | 0.595 | 0.604 | 0.130 | 0.425 | 0.510 |
| Title | 0.063 | 0.244 | 0.287 | 0.199 | 0.584 | 0.654 | 0.133 | 0.424 | 0.551 |
| Title and Abstract | **0.095** | **0.351** | **0.363** | **0.268** | **0.720** | **0.765** | **0.194** | **0.585** | **0.691** |

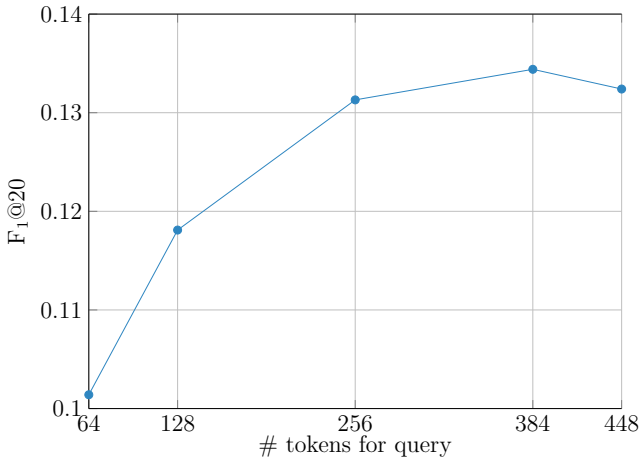


**Fig. 1.** $F_1$@20 on the development set when varying the number of tokens allocated to the input sequence (whose limit is 512 tokens) for the query (as opposed to the candidate document). The query is the concatenation of the title and abstract.

One of the limitations of transformer-based models (including BERT) is that memory consumption increases quadratically with the number of tokens in the input sequence. On modern hardware such as TPU v3s or GPU V100s, the maximum number of tokens that we can efficiently train a BERT-Large model is approximately 512. In our task, since the concatenation of query and candidate tokens is typically longer than this limit, there is a trade-off between the number of tokens we allocate to each sequence.

In Figure 1, we show how effectiveness changes as we allocate more tokens to the query than to the candidate document while limiting the sum of the two sequences to 512 tokens. These results are obtained with BM25 + SciBERT-Base (for faster experimental turnaround). The curve shows that query terms are more important to the re-ranker model, as increasing query tokens from 64 to 256 increases $F_1$@20 by 2 points. Decreasing candidate document tokens from 256 to 64 barely changes $F_1$@20. This result is somewhat surprising as one expects the two sequences to have equal importance in the task of query–document relevance estimation. Note that in all previous experiments (Table 2),

we used 256 tokens for the query and 256 for the candidate; this suggests that our main results might be even higher had we tuned this hyperparameter as well. Future work should investigate if this is particular to citation recommendation, or if it also occurs in other retrieval tasks with long queries as well.

## 6 Conclusions

We provide an extensive evaluation of pretrained transformer models for the scientific literature recommendation task. We find that in-domain pretraining and domain-specific vocabulary greatly improve effectiveness. Additionally, we present an unexpected finding: Despite the symmetry of the two inputs when trying to estimate the relevance of a candidate article to a query article, we find that terms from the query article are more important than terms from the candidate article in allocating "space" for BERT input. Future work should investigate this observation in more detail.

## Acknowledgments

This research was supported in part by the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, NVIDIA, and eBay. Additionally, we would like to thank Google for computational resources in the form of Google Cloud credits.

## References

1. Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., Kinney, R., Kohlmeier, S., Lo, K., Murray, T., Ooi, H.H., Peters, M., Power, J., Skjonsberg, S., Wang, L., Wilhelm, C., Yuan, Z., van Zuylen, M., Etzioni, O.: Construction of the literature graph in Semantic Scholar. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers). pp. 84–91 (2018)
2. Basu, C., Hirsh, H., Cohen, W.W., Nevill-Manning, C.: Technical paper recommendation: A study in combining multiple information sources. Journal of Artificial Intelligence Research **14**, 231–252 (2001)
3. Beltagy, I., Lo, K., Cohan, A.: SciBERT: Pretrained contextualized embeddings for scientific text. arXiv:1903.10676 (2019)
4. Bhagavatula, C., Feldman, S., Power, R., Ammar, W.: Content-based citation recommendation. arXiv:1802.08301 (2018)
5. Bollacker, K.D., Lawrence, S., Giles, C.L.: A system for automatic personalized tracking of scientific literature on the web. In: Proceedings of the Fourth ACM conference on Digital Libraries (DL '99). pp. 105–113 (1999)
6. Chen, T.T., Lee, M.: Research paper recommender systems on big scholarly data. In: Pacific Rim Knowledge Acquisition Workshop. pp. 251–260 (2018)

7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186 (2019)
8. Dinh, D., Tamine, L.: Combining global and local semantic contexts for improving biomedical information retrieval. In: European Conference on Information Retrieval. pp. 375–386 (2011)
9. Eto, M.: Extended co-citation search: Graph-based document retrieval on a co-citation network containing citation context information. Information Processing & Management **56**(6), 102046 (2019)
10. Fiorini, N., Canese, K., Starchenko, G., Kireev, E., Kim, W., Miller, V., Osipov, M., Kholodov, M., Ismagilov, R., Mohan, S., Ostell, J., Lu, Z.: Best Match: New relevance search for PubMed. PLoS Biology **16**(8), e2005343 (2018)
11. Fiorini, N., Leaman, R., Lipman, D.J., Lu, Z.: How user intelligence is improving PubMed. Nature Biotechnology **36**(10), 937 (2018)
12. Gao, Y., Kinoshita, J., Wu, E., Miller, E., Lee, R., Seaborne, A., Cayzer, S., Clark, T.: Swan: A distributed knowledge infrastructure for Alzheimer disease research. Web Semantics: Science, Services and Agents on the World Wide Web **4**(3), 222–228 (2006)
13. Ginsparg, P.: First steps towards electronic research communication. Computers in Physics **8**(4), 390–396 (1994)
14. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, C.L.: Context-aware citation recommendation. In: Proceedings of the 19th International Conference on World Wide Web. pp. 421–430 (2010)
15. Huang, W., Kataria, S., Caragea, C., Mitra, P., Giles, C.L., Rokach, L.: Recommending citations: Translating papers into references. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12). pp. 1910–1914 (2012)
16. Huang, W., Wu, Z., Liang, C., Mitra, P., Giles, C.L.: A neural probabilistic model for context based citation recommendation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
17. Jerome, R.N., Giuse, N.B., Gish, K.W., Sathe, N.A., Dietrich, M.S.: Information needs of clinical teams: Analysis of questions received by the clinical informatics consult service. Bulletin of the Medical Library Association **89**(2), 177 (2001)
18. Johnson, R., Watkinson, A., Mabe, M.: The STM report: An overview of scientific and scholarly publishing. International Association of Scientific, Technical and Medical Publishers (2018)
19. Kanakia, A., Shen, Z., Eide, D., Wang, K.: A scalable hybrid research paper recommender system for Microsoft Academic. In: The World Wide Web Conference. pp. 2893–2899 (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
21. Kodakateri Pudhiyaveetil, A., Gauch, S., Luong, H., Eno, J.: Conceptual recommender system for CiteSeerX. In: Proceedings of the Third ACM Conference on Recommender Systems. pp. 241–244 (2009)
22. Lawrence, S., Bollacker, K., Giles, C.L.: Indexing and retrieval of scientific literature. In: Proceedings of the 8th ACM International Conference on Information and Knowledge Management (CIKM '99). pp. 139–146 (1999)

23. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: A pre-trained biomedical language representation model for biomedical text mining. arXiv:1901.08746 (2019)
24. Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T.M., Xia, F.: Context-based collaborative filtering for citation recommendation. IEEE Access **3**, 1695–1703 (2015)
25. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019)
26. Lu, Y., He, J., Shan, D., Yan, H.: Recommending citations with translation model. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11). pp. 2017–2020 (2011)
27. McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work. pp. 116–125 (2002)
28. Mohan, S., Fiorini, N., Kim, S., Lu, Z.: Deep learning for biomedical information retrieval: Learning textual relevance from click logs. In: BioNLP 2017. pp. 222–231 (2017)
29. Nabeel Asim, M., Wasim, M., Usman Ghani Khan, M., Mahmood, W.: Improved biomedical term selection in pseudo relevance feedback. Database **2018** (2018)
30. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv:1901.04085 (2019)
31. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. arXiv:1906.05474 (2019)
32. Ren, X., Liu, J., Yu, X., Khandelwal, U., Gu, Q., Wang, L., Han, J.: ClusCite: Effective citation recommendation by information network-based clustering. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 821–830 (2014)
33. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proceedings of the 3rd Text REtrieval Conference (TREC-3). pp. 109–126. Gaithersburg, Maryland (1994)
34. Spangler, S., Wilkins, A.D., Bachman, B.J., Nagarajan, M., Dayaram, T., Haas, P.J., Regenbogen, S., Pickering, C.R., Comer, A., Myers, J.N., Stanoi, I.R., Kato, L., Lelescu, A., Labrie, J.J., Parikh, N., Lisewski, A.M., Donehower, L., Chen, Y., Lichtarge, O.: Automated hypothesis generation based on mining scientific literature. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1877–1886 (2014)
35. Sybrandt, J., Shtutman, M., Safro, I.: Moliere: Automatic biomedical hypothesis generation system. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1633–1642 (2017)
36. Yang, P., Fang, H., Lin, J.: Anserini: Enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). pp. 1253–1256 (2017)
37. Yang, P., Fang, H., Lin, J.: Anserini: Reproducible ranking baselines using Lucene. Journal of Data and Information Quality **10**(4), Article 16 (2018)