An Improved Test Collection and Baselines for Bibliographic Citation Recommendation

Dwaipayan Roy*
Indian Statistical Institute, Kolkata, India
IR Lab, CVPR Unit
dwaipayan_r@isical.ac.in

ABSTRACT

The problem of recommending bibliographic citations to an author who is writing an article has been well-studied. However, different researchers have used different datasets to evaluate proposed techniques, and have sometimes reported contradictory findings regarding the relative effectiveness of various approaches. In addition, these datasets are problematic in one way or another (e.g., in terms of size or availability), precluding the possibility of adopting one (or some) of them as standard benchmarks. A recently created test collection that makes use of data from CiteSeer^{x} is large, heterogenous, and publicly available, but has certain other limitations. In this paper, we propose a way to modify this test collection to address these limitations. We also use the improved test collection to establish a set of baseline results using elementary content-based techniques, as well as reference directed indexing.

CCS CONCEPTS

• **Recommender System → Bibliographic Citation Recommendation**;

KEYWORDS

bibliographic citations; recommender systems; test collections;

1 INTRODUCTION

Citing related work is an essential part of writing academic articles. As the number of publication venues increases across disciplines, it becomes more and more difficult to stay abreast of all developments in a particular area, especially in sub-disciplines that are only peripherally related to one's own. Many authors would, therefore, find a *bibliographic recommender system* (BRS) very useful. A general-purpose BRS may provide a wide range of functionalities: it may provide notifications (or "alerts") regarding new publications that are related to one's interests; it may also serve as an academic search engine. While BRSs may be fairly broad in their scope, for the purposes of this article, we define a BRS as a plugin-like tool that is integrated into an editor or word-processor used for writing academic articles. The tool monitors the article being written, and recommends references that may be cited at specific locations in the paper. In particular, the BRS should recommend citations whenever a citation placeholder (e.g., an opening square bracket) is typed. A BRS will generally recommend citations based on some or all of

- the text of the partially written paper (particularly the text in the vicinity of the citation's location),
- papers cited so far, and
- any prior literature searches performed by the author.

The CiteSight system [7] is a well-known example of such a tool. Since such tools are obviously useful (particularly to beginning researchers), the problem of designing a recommender system for bibliographic citations has been vigorously explored. In a recent and comprehensive survey, Beel et al. [1] report that there are more than 200 publications on this topic, with many of them appearing in the last 2–3 years.

Unfortunately, it is difficult to draw reliable conclusions about best practices from this large body of work. There is a good deal of variability in findings, with conflicting reports published (often by the same research group) about the relative efficacy of various methods and settings [1]. Arguably, the most important reason for this is the absence of standard datasets that may be used as benchmarks for evaluation. Indeed, in their survey, Beel et al. [1] conclude that it is "safe to say that no two studies, performed by different authors, used the same dataset." Moreover, the various datasets that have hitherto been used to evaluate BRSs all have significant drawbacks. Thus, it would be inadvisable to simply adopt one or more of these as benchmarks.

In sum, a large and heterogenous test collection that can serve as a benchmark for evaluating BRSs is urgently needed. Attempts have been made to create such a collection [14] by leveraging the publicly available CiteSeer^{x} dataset [3] (abbreviated as CSX henceforth). However, the test collection described in [14] also has certain shortcomings. Our goal in this article is to present an improved version of that test collection that is publicly available[1], along with quantitative results obtained on this dataset by using some well-known citation recommendation approaches.

2 RELATED WORK

In this section, we discuss research work related to BRSs from two points of view. We briefly consider techniques that have been used for generating bibliographic citation recommendations. Next, we summarise problems with the datasets that have been used for evaluation in well-known work on BRSs.

*It is a joint work with Mandar Mitra, Indian Statistical Institute, Kolkata. The author name can not be included due to publisher's policy.

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133085

[1]From http://www.isical.ac.in/~irlab/bcr.html.

Year	Dataset	# docs	# queries	# citations	Remarks
2006	[12]	4,200	151		Too small, too focused
2007	[15]	105,601	1000	1.46×10^6	Not available
2008	[13]	9,793	82	> 20,000	Too small, too focused
2010	[16]	597	28 researchers		Too small
2011	[8]	5,183	200	6,166	Too small, too focused; no response from authors regarding availability
2012	[6] (CiteSeer)	3,312	5-fold cross validation	26,597	Too small
2012	[6] (CiteULike)	14,418	5-fold cross validation	40,720	Too small; not available
2013	[17]	100,531	50 researchers		Docs., queries available as vectors only; no full-text / metadata
2014	[7]	2.3 million	1000		Proprietary, not available for general use
2016	[14]	630,351	2826	2,073,120	

Table 1: Sizes and drawbacks of various test collections used to evaluate BRSs. Not all figures were available for all collections. For the *too focused* collections, all papers are related to text processing, a small sub-domain of Computer Science.

2.1 Techniques

Recall from Section 1 that a BRS is expected to recommend citations based on some combination of the text of the partially written paper, the list of papers cited so far, and any prior literature searches performed by the author. In the following discussion, we use t, p, and s to denote these inputs to a BRS.

The two most popular approaches used by Recommender Systems are *Content-based methods*, and *Collaborative filtering* (CF) [10]. In the context of BRSs, a content-based method is one that recommends references by computing the similarity between t (a part of the *citing* paper's content), and the textual content of the target references (or *cited* papers). This similarity may be measured using standard IR techniques that compute vocabulary overlap, or using other, more sophisticated approaches [6, 8, 15]. In contrast, a CF based method generates recommendations based on p. The majority of BRSs proposed so far appear to use content-based methods, but CF has also been applied to this problem [2, 9]. We refer the reader to [1] for a comprehensive survey, and focus here only on *reference-directed indexing* (RDI), a technique that has been well-studied by Ritchie et al. in a series of papers [11–13]. The main idea behind RDI is to represent the content of a cited document using terms from citing documents, instead of terms contained in the cited document itself. More specifically, a cited document is indexed by terms that occur in the neighbourhood of its citations within citing documents. In a sense, RDI is a content-based technique that also uses some ideas from collaborative filtering.

2.2 Test Collections

The task of a BRS (as defined in Section 1) may be viewed as a variant of the ad hoc IR task [5]. Thus, a test collection for evaluating such BRSs will consist of the following components.

- A document collection containing a large number of scholarly articles on diverse topics, from which the BRS is expected to retrieve recommendations.
- A set of user queries. As discussed in Section 2.1, a query will generally consist of some combination of t, p, and s.
- Relevance assessments, i.e., a list of references that are deemed to be relevant at a particular location in the paper.

In [1], the authors discuss a number of test collections that have been used in the past by researchers, and point out the drawbacks of these collections. Following their findings, in Table 1, we have summarised some of the most promising test collections that have been used in the past for evaluation in well-known work on BRSs. Only the main drawbacks of these datasets are summarised in the last column of Table 1. For a detailed discussion of these collections, please refer to [1, 14].

Compared to standard IR datasets provided by TREC (http://trec.nist.gov), most of the collections are much too small to be used as realistic benchmarks for evaluating BRS systems. The Rexa database used in [15] is reasonably large, but like CiteSeer [4], it is a dynamic collection that grows over time, and the actual snapshot used for experiments in [15] no longer seems to be available. The dataset created by Sugiyama et al. [17, 18] is also moderately large and readily available, but it only provides documents and queries as TF-IDF weighted vectors, thus severely limiting the range of experiments that can be conducted using this dataset. The data used by the CiteSight system [7] is most promising in terms of size and heterogeneity. Unfortunately, it consists of papers provided by Microsoft Academic, and appears not to be available for general use.

3 PROPOSED COLLECTION

As discussed in Section 2.2, a test collection consists of (1) a document collection, (2) a set of queries, and (3) relevance judgement. In this section, we describe these components of the proposed collection.

CSX[3] seems to be a promising starting point for constructing a test collection for BRSs. This dataset is a carefully processed subset of a snapshot of the CiteSeer repository [4]. It consists of 630,351 XML files, with each file corresponding to one article. The files contain automatically extracted metadata (authors, title, abstract, etc.) in addition to extracts from the full text of the corresponding papers. The full text extracts consist of a series of *citation contexts*.

A citation context is defined as the textual content that surrounds a citation in the body of a paper. For each citation context, a unique numeric identifier of the cited paper is also provided.

The articles in CSX cover diverse topics related to Computer Science. Thus, this collection is fairly large and heterogenous. Being a static collection, it is usable for conducting reproducible experiments. Of course, CSX by itself is only a *document* collection. It has to be supplemented by a set of search queries and relevance judgments in order to convert it into a test collection.

As an initial attempt at creating a test collection out of CSX, the authors in [14] adopted a commonly used approach [7, 8]. A citation context can be taken as a query; the cited reference(s) would then be regarded as the relevant document(s) for that query. From the whole collection of documents, 226 papers were selected as *query* papers. The distinct contexts from these papers were extracted to form the actual queries. A total of 2,826 queries were thus obtained.

After some preliminary experiments on this collection, the following problems [14] were observed.

(1) In CSX, the citation context is defined as a fixed-size window of exactly 400 characters with the citation at its centre. As a result, contexts frequently begin and end in mid-word.
(2) If a reference is cited multiple times in a paper at different locations, the corresponding contexts, extracted from different parts of the paper, are simply concatenated. Such artificially created contexts would not naturally arise in the use-case discussed in Section 1. The corresponding queries are therefore unrealistic.
(3) If n references are cited in a single context but in different places (e.g., citations 1, 3, 9, 22 in Figure 1), only the middle most citation placeholder (22 in Figure 1) is considered as a relevant citation for that context. Thus, a system gets no credit at all for retrieving citations 1, 3 or 9 in response to this query. This is counter-intuitive.

We propose to address the above problems by using natural paragraphs instead of 400 character windows as citation contexts. Queries could continue to be created using this modified notion of a context. *All* references cited within the query paragraph would be regarded as relevant documents.

While this simple modification addresses all 3 limitations listed above and leads to more realistic queries, it cannot directly be applied to CSX, since CSX only contains 400 character windows, rather than the full text or even full paragraphs. Luckily, a full-text dump of the original CiteSeer data (> 1 million papers) is readily available on request. However, the full text files in CiteSeer appear to have been generated automatically from PDF files without any markup, and are considerably noisy. Thus, in this work, the improved dataset is created using the CiteSeer data, in combination with the metadata provided by CSX.

The CiteSeer dump that we obtained had 630,199 papers in common with CSX. Among these papers, 50 papers were manually selected as query papers. These papers comprise journal, conference, and workshop publications. The number of references cited by these papers ranges from 17 to 132 with the median being 35. From these 50 papers, natural paragraphs were selected and extracted manually. Paragraphs with multiple citations were chosen as queries, which results in multiple relevant papers per query. Also, the paragraphs were chosen carefully from all parts of the paper,

```
<raw>Nath, S., Gibbons, P. B., Seshan, S.,  ...
<contexts>a specific probabilistic counting scheme, and a discussion of
probabilistic counting schemes trade-offs and limitations. Probabilistic
counting selects several representative elements, or a synopsis =-=[22]-=-,
as an estimator for the total number of distinct elements [1, 3, 9]. The
synopsis summarizes the entire element set and thus permits estimation of
the total size. Probabilistic counting provides a t mallest observed
element, and e0 and e1 the minimal and maximal value, respectively.
Generally a probabilistic counting scheme provides three functions on
synopses: Generation, Fusion, and Evaluation =-=[22]-=-. A Generation
function selects the representative items from the input set I to use
as a synopsis S. In this paper, we consider a class of probabilistic
counting schemes whose usion function prevent</contexts>
<clusterid>44856</clusterid>
```

Figure 1: A context for which only one reference is counter-intuitively regarded as relevant (taken from [14])

to ensure that their content is focused to a particular subject (and not ambiguous). A total of 171 query paragraphs were created. The unique numeric ids for the relevant documents (i.e., the citations) for each query were extracted using CSX metadata. To summarise, the three components of the proposed collection are the following.

Document Collection. A total of 630,149 papers (= 630,199 − 50 query papers) both in the form of meta-content (obtained from CSX) as well as full-content (from CiteSeer).

Query Set. 50 papers chosen manually from the whole collection as query papers. From these 50 papers, a total of 171 paragraphs were chosen as queries [2]. The types of these query papers are graphically presented in Figure 2.

Relevance Judgement. The papers cited in a query para constitute the set of relevant documents for that query. Their numeric ids are extracted from the CSX meta data of the query papers.

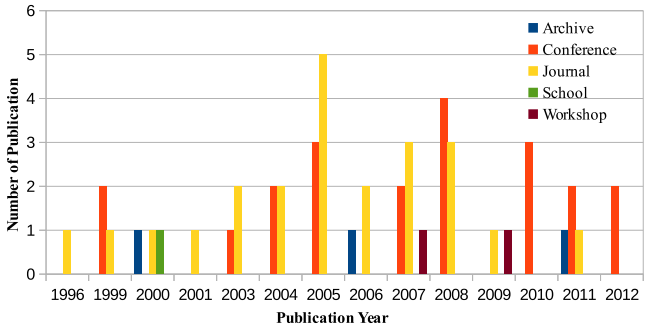

Figure 2: The types of the selected papers, that are used as queries in the proposed collection.

In the next section, we present some baseline results obtained using content-based techniques. Note, however, that this dataset can also be used for experimenting with CF-based techniques, since the citation graphs for CSX and CiteSeer are available.

4 BASELINE RESULTS

We used content-based techniques to obtain a set of baseline figures for this collection. Documents and queries were indexed using Lucene.[3] SMART Stopword list and Porter's stemmer were used during preprocessing. As mentioned in Section 3, the title and abstract of each paper is clearly marked up in CSX. Thus, a number of different combinations of document fields can be searched during

[2] A sample query: http://www.isical.ac.in/~irlab/data/bcr/sample-query.xml
[3] https://lucene.apache.org/

Config.	TFIDF	BM25	JM	D
D-content	0.0794, 211	0.1012, 205	0.0946, 220	0.0974, 221
Q-text		$k1 = 1.8, b = 0.75$	$\lambda = 0.7$	$\mu = 1500$
D-content	0.0783, 211	0.0851, 208	0.0942, 219	0.0970, 223
Q-text+title		$k1 = 1.8, b = 1$	$\lambda = 0.5$	$\mu = 100$
D-title+abst.	0.0728, 175	0.0824, 172	0.0787, 171	0.0755, 173
Q-text		$k1 = 1.0, b = 0.7$	$\lambda = 0.5$	$\mu = 100$

Table 2: Direct content based methods: best results (MAP, number of relevant documents retrieved). When the full content of documents is used for retrieval (first and second columns), the performance is significantly better (based on a paired t-test at the 5% level) than the performance when only the title and abstract are used.

retrieval. Similarly, we may also think of supplementing each query paragraph with the title and abstract of the corresponding query paper, in order to provide some additional context for the query paragraph. We tried the following plausible configurations in our experiments [4]:

- **D-content-Q-text**: full text of documents searched using only query paragraph;
- **D-content-Q-text+title**: full text of documents searched using query paper title along with query paragraph;
- **D-title+abstract-Q-text**: only the title and abstract of documents searched using query paragraph.

For each query, 100 papers were retrieved using the following models: (1) TF-IDF, (2) BM25, and Language Modeling with (3) Jelinek Mercer and (4) Dirichlet smoothing. For BM25, $k1$ is varied from 1.0 to 2.0 in the steps of 0.1, and b is varied from 0 to 1 in steps of 0.25. The smoothing parameters of the language models λ (Jelinek Mercer), and μ (Dirichlet) are respectively varied in steps of 0.1 in between (0.1 - 0.9), and over the range $\{100, 200, 500, 1000, 2000, 2500, 3000, 5000\}$. The results obtained, along with the parameter setting that gives the best MAP, are reported in Table 2.

Config.	TFIDF	BM25	JM	D
Q-text	0.2887, 373	0.2900, 365	0.2921, 380	0.2886, 373
		$k1 = 1.2, b = 0.4$	$\lambda = 0.2$	$\mu = 700$
Q-text+title	0.2852, 374	0.2845, 380	0.2931, 384	0.2854, 377
		$k1 = 1.6, b = 0.5$	$\lambda = 0.3$	$\mu = 500$

Table 3: RDI: best results (MAP, no. of relevant documents retrieved). All RDI runs are significantly better (based on paired t-test the 5% level) than the runs reported in Table 2.

Reference Directed Indexing

Recall that in RDI, a document D is indexed using terms from citation contexts in other documents that cite D. Since CSX provides precisely this information, we used CSX to create a reference directed index for our collection. Note that the configurations listed above do not make sense for RDI. For experiments involving RDI, we simply tried using either the query paragraph alone, or in combination with the title of the query paper. The same retrieval models as listed above were used. Results are reported in Table 3. The following conclusions may be drawn from Tables 2 and 3.

- RDI works much better than direct content-based indexing.
- Indexing the full text of documents is better than indexing the title and abstract only.
- Adding the title of the query paper to the query paragraph does not seem to have much impact.

[4]Source code available from: https://github.com/dwaipayanroy/citereco

5 CONCLUSION

The improved test collection described in this paper has significant advantages over similar datasets that have been used in earlier work. Since it is publicly available from *http://www.isical.ac.in/ irlab/bcr.html*, we hope that it can be used by different research groups to reliably compare different approaches to bibliographic citation recommendation. In future work, we plan to cover a more comprehensive set of baselines that includes such techniques.

6 ACKNOWLEDGMENTS

This is a joint work with Mandar Mitra, Indian Statistical Institute, Kolkata. He has significantly contributed to the work and on preparation of this draft.

REFERENCES

[1] Joeran Beel, Bela Gipp, and Corinna Breitinger. 2015. Research paper recommender systems – a literature survey. *International Journal on Digital Libraries* (2015). http://docear.org/papers/research_paper_recommender_systems_--_a_literature_survey.pdf

[2] Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra, and C. Lee Giles. 2013. Can't see the forest for the trees?: a citation recommendation system. In *Proc. 13th ACM/IEEE-CS JCDL*. ACM, 111–114. http://dl.acm.org/citation.cfm?id=2467743

[3] Cornelia Caragea, Jian Wu, Alina Ciobanu, Kyle Williams, Juan Fernandez-Ramirez, Hung-Hsuan Chen, Zhaohui Wu, and Lee Giles. 2014. CiteSeer x: A Scholarly Big Dataset. In *Advances in Information Retrieval*. Springer, 311–322.

[4] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. 1998. CiteSeer: An automatic citation indexing system. In *Proc. 3rd ACM conf. Digital Libraries*. ACM, 89–98.

[5] Donna Harman. 2011. *Information Retrieval Evaluation*. Morgan & Claypool Publishers. DOI : http://dx.doi.org/10.2200/S00368ED1V01Y201105ICR019

[6] Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C. Lee Giles, and Lior Rokach. 2012. Recommending citations: translating papers into references. In *Proc. CIKM*. ACM, 1910–1914. http://dl.acm.org/citation.cfm?id=2398542

[7] Avishay Livne, Vivek Gokuladas, Jaime Teevan, Susan T. Dumais, and Eytan Adar. 2014. CiteSight: supporting contextual citation recommendation using differential search. In *Proc. SIGIR*. ACM Press, 807–816.

[8] Yang Lu, Jing He, Dongdong Shan, and Hongfei Yan. 2011. Recommending citations with translation model. In *Proc. CIKM*. ACM, 2017–2020.

[9] Sean M. McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. On the Recommending of Citations for Research Papers. In *Proc. of 2002 CSCW*. 116–125.

[10] F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor. 2011. *Recommender Systems Handbook*. Springer US.

[11] Anna Ritchie, Stephen Robertson, and Simone Teufel. 2008. Comparing citation contexts for information retrieval. In *Proc. CIKM*. ACM, 213–222.

[12] Anna Ritchie, Simone Teufel, and Stephen Robertson. 2006. Creating a test collection for citation-based IR experiments. In *Proc. HLT-NAACL*. ACL, 391–398.

[13] Anna Ritchie, Simone Teufel, and Stephen Robertson. 2008. Using terms from citations for IR: some first results. In *Advances in Information Retrieval*. 211–221.

[14] Dwaipayan Roy, Kunal Ray, and Mandar Mitra. 2016. From a Scholarly Big Dataset to a Test Collection for Bibliographic Citation Recommendation. In *Scholarly Big Data: AI Perspectives, Challenges, and Ideas - AAAI Workshop, 2016.*

[15] Trevor Strohman, W. Bruce Croft, and David Jensen. 2007. Recommending citations for academic papers. In *Proc. SIGIR*. ACM, 705–706.

[16] Kazunari Sugiyama and Min-Yen Kan. 2010. Scholarly Paper Recommendation via User's Recent Research Interests. In *Proc. 10th JCDL*. 29–38.

[17] Kazunari Sugiyama and Min-Yen Kan. 2013. Exploiting Potential Citation Papers in Scholarly Paper Recommendation. In *Proc. 13th JCDL*. 153–162.

[18] Kazunari Sugiyama and Min-Yen Kan. 2015. Scholarly Paper Recommendation Datasets (NUS). (2015). http://www.comp.nus.edu.sg/~sugiyama/SchPaperRecData.html