The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Scholarly Big Data: AI Perspectives, Challenges, and Ideas:
Technical Report WS-16-13

From a “Scholarly Big Dataset” to a Test Collection for Bibliographic Citation Recommendation

Dwaipayan Roy
IR Lab, CVPR Unit
Indian Statistical Institute, Kolkata

Kunal Ray
Microsoft IDC
Bangalore

Mandar Mitra
IR Lab, CVPR Unit
Indian Statistical Institute, Kolkata

Abstract

The problem of designing recommender systems for scholarly article citations has been actively researched with more than 200 publications appearing in the last two decades. In spite of this, no definitive results are available about what approaches work best. Arguably the most important reason for this lack of consensus is the dearth of standardised test collections and evaluation protocols, such as those provided by TREC-like forums. CiteSeer^X, a “scholarly big dataset” has recently become available. However, this collection provides only the raw material that is yet to be moulded into Cranfield style test collections. In this paper, we discuss the limitations of test collections used in earlier work, and describe how we used CiteSeer^X to design a test collection with a well-defined evaluation protocol. The collection consists of over 600,000 research papers and over 2,500 queries. We report some preliminary experimental results using this collection, which are indicative of the performance of elementary content-based techniques. These experiments also made us aware of some shortcomings of CiteSeer^X itself.

Introduction

A substantial amount of research has been done on recommender systems for bibliographic citations. A recent survey of research-paper recommender systems (Beel, Gipp, and Breitinger To appear) reports that more than 200 research articles have been published on this subject in the last two decades, with 30% of these appearing in the last two years. In spite of the volume of work done, the wide variety of approaches tried, and the continuing level of interest in this general research area, no definitive results are available about what methods work best. Beel, Gipp, and Breitinger (To appear) provide concrete examples of conflicting reports that have been published regarding the relative effectiveness of content-based and collaborative filtering based techniques. Possibly the most important reason for this lack of consensus is the dearth of standardised test collections and evaluation protocols, such as those provided by TREC-like forums. Thus, for a researcher starting out in this area, it is not at all clear what datasets she should use for evaluation, and what baseline results she should compare against. This situation is almost exactly like the one prevalent in the Text Classification community in the early 90s (Lewis 2004).

The recent availability of CiteSeer^X (Caragea et al. 2014), a “scholarly big dataset”, is a first step towards remedying this situation. This dataset is a cleaned subset of the complete data available from the CiteSeer project (Giles, Bollacker, and Lawrence 1998), one of the best known open access repositories of scholarly articles in Computer Science and related fields.[1] CiteSeer uses automatic metadata extraction methods which are reasonably accurate, but result in numerous errors nonetheless. One of the primary motivations behind CiteSeer^X appears to have been the creation of a well-structured dataset with clean, reliable metadata. This makes it far more usable than the original CiteSeer data for experimental research in bibliographic recommender systems (BRS). However, in its present form, CiteSeer^X is only a *document* collection. In order to create a *test* collection, it has to be supplemented by a set of search queries and relevance judgments (Ritchie, Teufel, and Robertson 2006).

Our goal in this article is two-fold. First, we argue that existing test collections that have been used in recent times for evaluating BRS have significant drawbacks. We believe that CiteSeer^X can serve as the starting point of a benchmark that is closer to the de facto standard in terms of size, heterogeneity and flexibility. Accordingly, our second goal is to design a test collection that uses CiteSeer^X, and has a well-defined evaluation protocol. The proposed collection consists of over 600,000 research papers and over 2,500 queries. This collection is intended to serve the same purpose that the “ModApte split” (Apté, Damerau, and Weiss 1994) served for the Reuters text classification collection. Next, we describe some preliminary experiments with this testbed which made us aware of some shortcomings of CiteSeer^X itself. We conclude by outlining the remedial measures that need to be taken in order to convert this dataset to a useful and reliable test collection.

Related work

As with recommender systems in general, the two major paradigms that have been proposed for BRS are collaborative filtering (CF) based approaches and content based approaches. A comprehensive survey of research paper recommender systems can be found in (Beel, Gipp, and Breitinger To appear). We focus here on a relatively small number of

[1] CiteSeer^X is actually the new incarnation of the original CiteSeer project. To avoid confusion, we refer to the complete dataset as CiteSeer, and use CiteSeer^X to refer to the cleaned subset.

recent papers that are most relevant to our current objective.

Collaborative filtering approaches. In a recent study, Caragea et al. (2013) applied singular value decomposition to the citation graph in an attempt 'to construct a latent "semantic" space, where citing and cited papers that are highly correlated are placed close to each other.' On a December 2011 snapshot of the CiteSeer citation graph, this approach outperforms the traditional CF approach (McNee et al. 2002) to generating recommended citations.

Content based methods. According to Beel, Gipp, and Breitinger (To appear), more than half the BRS surveyed used content-based approaches. As expected, the vocabulary mismatch between a citing and a cited paper poses a problem for content-based methods. In order to address this mismatch, recent efforts (Lu et al. 2011; Huang et al. 2012) have formulated citation recommendation as a translation task: the languages used in citing and cited papers are assumed to be different, and a statistical machine translation model is trained to "translate" a citation context to a citation. Surprisingly, Huang et al. (2012) report that a simple representation for the cited papers, where each paper is represented by a single, unique identifier, works better than when the cited papers are represented by their words as in (Lu et al. 2011).

Test collections for BRS

Bibliographic citation recommendation is a somewhat broad topic. Different variations of this general problem correspond to various intended use cases. To create a usable test collection for evaluating a BRS, we need to be aware of the specific use case that the BRS in question intends to address. In this study, we are interested in the following use case. While writing an article, a user needs precise bibliographic details of a reference that she wishes to cite at a specific point within the paper, e.g., in the form of a BIBTEX record. Two sub-cases may arise.

First, the user may know which article she wants to cite. The details for this article may already be available to her in a .bib file or similar database, or they may be available in online resources such as the ACM Digital Library, Google Scholar, or ScienceDirect. In such a situation, the user typically provides a string or a regular expression corresponding to some substring of the bibliographic record. The system shows matching records (fetched either from the current bibliographic database or an online resource). Since this scenario essentially involves string or regular expression matching, it is simple enough to be regarded as a solved problem. Indeed, there are tools available that serve this purpose quite effectively.

We would like to consider the second case in which the user either does not know the precise article(s) that she wants to cite, or is aware of some appropriate citations, but is looking for additional references. In such a situation, the citation recommendation task may be viewed as a form of the ad hoc IR search task (Harman 2011). Accordingly, a recommender system may be evaluated using a standard, Cranfield-like test collection consisting of the following (Cleverdon 1997).

- A document collection containing a sufficiently comprehensive set of scholarly articles.
- A set of user queries. A query might consist of one or more of the following: the text surrounding the citation; additional text from the paper (e.g., the abstract) that could help to establish a broader context for the words in the immediate vicinity of the citation; a partial list of the references already cited by the author; etc.
- Relevance assessments, i.e., a list of references that are deemed to be relevant at a particular location in the paper.

As suggested in the Introduction, it is far from clear what approach would be most effective for solving this problem: content-based approaches, collaborative filtering, some combination of these two, or something else altogether. In the rest of this section, we discuss, from this perspective, test collections that have been used over the last 10 years for evaluation of BRS. Table 1 provides a summary of the collections described here.

Ritchie et al.

Ritchie, Teufel, and Robertson (2006) created a test collection consisting of about 4,200 papers taken from the ACL Anthology[2]. The query set, consisting of 151 queries, was created by asking authors of papers that were accepted for ACL-2005 and HLT-EMNLP-2005 "to formulate the research question(s) behind their work and to judge how relevant each reference in their paper was to each of their research questions", on a 4-point scale. In later work, Ritchie, Teufel, and Robertson (2008) used a collection of 9,793 papers, also taken from the ACL Anthology, along with 82 queries that were created as before. For this collection, the query creators were asked to judge a pool of potentially relevant papers in addition to the references actually cited in their papers. This collection is still available on request from its creators. It is well-organised, and can easily be used for retrieval experiments. Its main drawback is that it is too small and homogeneous in nature (particularly in comparison to standard test collections currently in use for other information processing problems) to serve as a realistic testbed for the use case discussed above.

Sugiyama et al.

Sugiyama and Kan have also created two datasets[3] that are publicly available. The first dataset (Sugiyama and Kan 2010) consists of the following.

- The publications of 15 *junior* researchers (only one published paper from each), and 13 *senior* researchers (each having multiple publications). These publications are used to construct "interest profiles" for these 28 researchers.
- A list of 597 full papers, published in ACL during 2000–2006. These were considered as candidate papers to be recommended to the above researchers.
- A set of relevance judgements. The researchers involved were asked to judge whether each candidate paper was relevant / non-relevant to his / her research interests.

[2] https://aclweb.org/anthology

[3] www.comp.nus.edu.sg/~sugiyama/SchPaperRecData.html

Data	# documents	# queries	# citations
Ritchie, Teufel, and Robertson (2006)	4,200	151	
Ritchie, Teufel, and Robertson (2008)	9,793	82	$>$ 20,000
Sugiyama et al. (Sugiyama and Kan 2010)	597	28 researchers	
Lu et al. (Lu et al. 2011)	5,183	200	6,166
CiteSeer (Huang et al. 2012)	3,312	5-fold cross validation	26,597
CiteULike (Huang et al. 2012)	14,418	5-fold cross validation	40,720
Sugiyama et al. (Sugiyama and Kan 2013)	100,531	50 researchers	
CiteSeer^X	**630,351**	2826	**2,073,120**

Table 1: Sizes of various test collections used to evaluate BRS.

A second, substantially larger dataset (Sugiyama and Kan 2013) consists of 100,531 papers from various ACM proceedings published during 2000–2010, along with the recent publications of 50 researchers. All papers are provided in the form of feature vectors, rather than full text. Candidate papers are available as term vectors with TF-IDF weights, while the publications of the target researchers are provided as simple TF-weighted vectors. Relevance judgments were obtained as before.

The main drawback of these datasets from our perspective is that they address a different use case from the one that we are focusing on. Sugiyama and Kan's goal is to construct a personalised recommended-reading list for a user based on her research interests, rather than to recommend specific references appropriate for a particular location within a paper that is being written. An additional limitation is that the papers are only available as feature vectors. Neither the full text nor any metadata (e.g., title, authors, abstract, sections) is available for these papers. Thus, this dataset cannot be used to investigate techniques that make use of these sources of information. Indeed, even simply reproducing the findings in (Sugiyama and Kan 2013) appears to be impossible using their data, since the authors investigate which sections of papers can be leveraged to represent papers effectively, and conclude that the Conclusion is generally an important section for this purpose.

Lu et al.

Lu et al. (2011) address precisely the use case that we have in mind. The dataset used in their experiments "is a collection of 5,183 papers from 1988 to 2010, mainly in the information retrieval and text mining direction." About 1,500 papers out of these 5,183 papers have been cited by the other papers in the collection. A total of 6,166 citations refer to papers in the collection. Two hundred of these citations were randomly selected along with their *contexts* (a citation context is defined as the three sentences surrounding a citation [sic]) and used as queries. Unfortunately, this collection does not seem to be available either publicly or on request. In any case, this dataset is also small and homogeneous, like those created by Ritchie et al.

CiteSeer

The same use case is also studied in Huang et al. (2012). For evaluation, they used a dataset consisting of 3,312 papers taken from CiteSeer. This dataset is available[4], but once again, it is small, and contains publications from a small sub-domain of Computer Science. Moreover, each paper is described by a term vector with binary weights signifying only the absence / presence of the word in the paper). This severely limits the scope of experiments that can be conducted using this data. Finally, the dataset mentions a *dictionary* that presumably provides a mapping from words to the dimensions of the vector space, but this dictionary is missing from the tarball that we downloaded. Thus, we are only left with a bag-of-*unknown*-words representation of the papers.

CiteULike

Along with the Citeseer dataset mentioned above, Huang et al. (2012) also used a snapshot of CiteULike from November 2005 to January 2008 containing 14,418 papers. This dataset also suffers from the now familiar limitations of being both small in size and unavailable.

Creating the test collection

CiteSeer^X is large, heterogeneous, contains (partial) full-text as well as metadata. Thus, it seems to be the most promising starting point for constructing a realistic test collection for our purpose. We start by presenting a brief overview of CiteSeer^X (Caragea et al. 2014) as a document collection.

CiteSeer^X

The collection consists of 630,351 XML files. Each file corresponds to one article, and is identified by its DOI. The articles are drawn from a variety of disciplines related to Computer Science (Algorithms, Artificial Intelligence, Databases, Networking, Security, Vision, etc.) as well as Mathematics and Statistics. Metadata such as the title, abstract, the name of each individual author, and the publication venue and year are marked up using appropriate tags. The full-text is not provided, even though it is available separately via http://csxstatic.ist.psu.edu/about/data. Instead, the main body of each file comprises a series of citations, each consisting of the "raw" reference extracted from the bibliography, the *citation context* (the textual content in the body of a paper that surrounds a citation), and a *clusterid*. The actual citation (in numerical, author/year, and various other

[4] http://linqs.umiacs.umd.edu/projects//projects/lbc/index.html

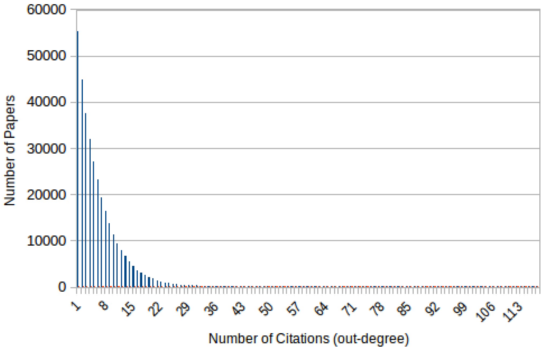


Figure 1: Distribution of no. of citations (out-degree) in CiteSeer^X

formats) is marked by delimiters (=−= and −=−). Since the articles in the CiteSeer^X repository have been automatically crawled and processed, often, multiple versions of the same paper are present in the database. Each version of each article gets a unique DOI, but the different versions are clustered together and assigned the same clusterid. For papers not included in the repository, a clusterid of 0 is assigned.

Queries and relevance judgments

To construct a test collection based on CiteSeer^X, we adopt the approach used by Lu et al. (2011). The textual part of a citation context forms a query, and the cited references are taken to be the relevant documents for that query.

We use a simplified form of stratified sampling to randomly select a set of 226 papers from the distribution showed in Figure 1 as *query* papers. The distinct contexts from these papers were taken to be the actual queries. The title and abstract of the query paper were also included as additional fields. A total of 2,826 queries were thus obtained.

Most queries (contexts) have only one relevant citation, but a few have more (Figure 3). In CiteSeer^X, if n references are cited together, the context is repeated n times, once corresponding to each reference. For example, the context shown in Figure 2 is repeated for $4, 5$ and 18 in the corresponding file. We regard these repeated contexts as a single query, and the n references are counted as n relevant documents for this query.

The assumption that the references cited in a context are the only relevant documents for the corresponding query makes it possible to create a large test collection from a source like CiteSeer^X without any human assessment effort, but of course, it is possible that there are other references that could also be regarded as relevant for that context.

Note that the citation graph for both CiteSeer^X and CiteSeer are either available, or can be constructed. Thus, the dataset can be used for experimenting with, and fairly comparing, both CF- and content-based techniques that have been proposed earlier.

```
<raw>Feng W, Jahanian F, and Sechrest S Providing VCR ...
<contexts>ponding video frame in the fast#forward
\Thetale. The mapping can also add delay to the VCR
operations. We note that although band# width
renegotiation is another way to provide the VCR
functionality =-=[4, 5, 18]-=-, this approach is not
suitable for a limited#bandwidth environment. We recall
that the band# width is not even enough to support the
normal playback. To address the \Thetarst problem, one
must be abl
</contexts>
<clusterid>258307</clusterid>
```

Figure 2: An example citation from the CiteSeer^X collection with three references cited together

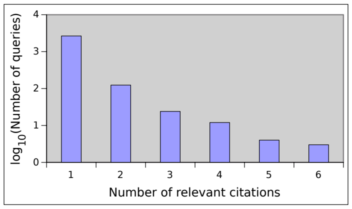


Figure 3: Number of queries vs. number of relevant citations

Baseline results

As the obvious next step, we used some elementary content-based techniques to obtain a set of baseline figures for this test collection. We used Lucene[5] for our experiments. Stopwords were removed using the Smart stopword list[6], and words were stemmed using Porter's stemmer (Porter 1997). For each query, 100 documents were retrieved. using the following IR models in turn: (i) Lucene's built-in TF-IDF vector based ranking scheme, (ii) the probabilistic model BM25 (Robertson and Zaragoza 2009), and (iii) language modeling (LM) (Hiemstra 2001; Ponte and Croft 1998) with both Jelinek-Mercer (JM) and Dirichlet (D) smoothing (Zhai and Lafferty 2001). A range of parameter values were tried for models (ii) and (iii).

Query and document fields

We experimented with indexing three combinations of query and document fields.

1. **Dtac-Qtac**: All three fields (**t**itle, **a**bstract, **c**ontext) are indexed for both **D**ocuments and **Q**ueries.
2. **Dtac-Qc**: As above, but the abstract and title of the query paper are ignored when indexing the query context.
3. **Dta-Qc**: As above, but the contexts of documents are not indexed. This setting corresponds to a situation where the titles and abstracts of papers are available, but the full-text (of which the contexts form a part) is unavailable.

Results

The best results in terms of Mean Average Precision (MAP) obtained using these methods are summarised in Table 2.

[5] https://lucene.apache.org/

[6] ftp://ftp.cs.cornell.edu/pub/smart/

Configuration	TFIDF	BM25	JM	D
Dtac-Qtac	0.0914, 1251	0.1012, 1326 $k1 = 1.2, b = 1.0$	0.0971, 1299 $\lambda = 0.4$	0.0957, 1289 $\mu = 100$
Dtac-Qc	0.1695, 1597	0.1793, 1668 $k1 = 1.2, b = 0.7$	0.1782, 1667 $\lambda = 0.4$	0.1723, 1653 $\mu = 100$
Dta-Qc	0.1694, 1487	0.1727, 1505 $k1 = 1.2, b = 0.7$	0.1669, 1444 $\lambda = 0.2$	0.1695, 1492 $\mu = 100$

Table 2: Best results (MAP, number of relevant documents retrieved) using various retrieval models

The parameter settings at which the best results are obtained are also indicated. From the results, we can draw the following unsurprising conclusions:

- On this collection, simple content-based methods are not as bad as reported by Strohman, Croft, and Jensen (2007), but it is true that they do not work well.
- A focused context-only query seems to work better than including keywords from the title and abstract in the query.
- The availability of (parts of) the full-text of articles is generally helpful.

Limitations

A preliminary post-mortem of the results reveals that this dataset, despite definite advantages over the datasets discussed before, also suffers from certain limitations.

1. In CiteSeer^X, the citation context is defined as a fixed-size window of 400 characters with the citation at its centre. As a result, contexts frequently begin and end in mid-word (see Figures 2, 4 and 5). This also limits the scope of experiments involving contexts of varying size (such as those reported in (Ritchie, Robertson, and Teufel 2008)).
2. The actual citations (in numerical, author/year, and various other formats) are marked by delimiters (=−= and −=−). However, because unicode characters have not always been handled correctly, these delimiters have, in many cases, been inserted a few byte positions away from their intended position (see Figure 4). To correct these errors, we counted the number of non-ASCII characters occurring in a context, and adjusted the positions of the delimiters accordingly. A random check suggests that this "post-processing" step corrected most errors of this type.
3. If a reference is cited multiple times in a paper at different locations (e.g. citation 22 in Figure 5), the corresponding contexts are simply concatenated without any separators marking the boundary between two contexts. In this case also, a straightforward post-processing step may be used to insert a delimiter between concatenated contexts.
4. If n references are cited in a single context but in different places (e.g. citations 1, 3, 9, 22 in Figure 5), only the middle most citation placeholder (22 in Figure 5) 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.

```
<raw>E. Airoldi, ... </raw>
<contexts>o deal with collusion, entities can compute
reputation subjectively, where player A weighs player
B's opinions based on how much player A trusts player
B. Our subjective algorithm is based on
maxflo=-=w [24] [32-=-]. Maxflow is a graph theoretic
problem, which given a directed graph with weighted
edges asks what is the greatest rate at which
''material'' can be shipped from the source to the
target without vi </context>
```

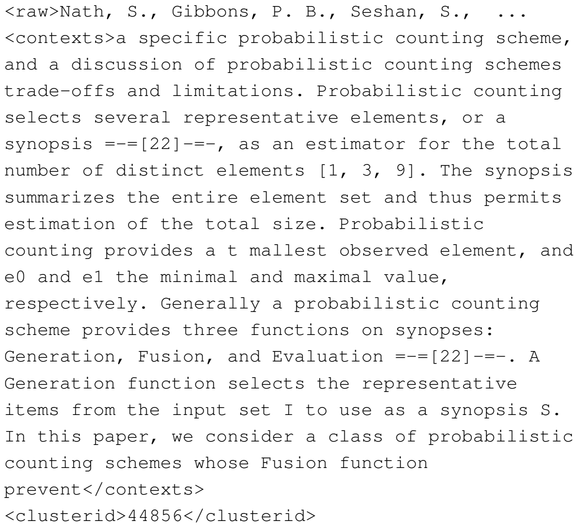
Figure 4: An example citation from the CiteSeer^X collection with misplaced delimiters due to presence of Unicode

Discussion

From an evaluation perspective, Limitation 4 discussed above is possibly the most important. Considering natural paragraphs instead of 400 character windows as citation contexts could be one possible way to address this problem. Queries could continue to be created using this modified notion of a context. *All* references cited within the paragraph would be regarded as relevant documents. Of course, this approach requires access to the full text of the papers. Luckily, this is available as part of the original CiteSeer data. However, the full texts in CiteSeer appear to have been generated automatically from PDF files, and are noisy. Nevertheless, it should be possible to create a query collection using only a small amount of manual effort by combining CiteSeer and CiteSeer^X. We have started constructing a modified test collection based on this idea, and hope to make the modified dataset publicly available in the near future.

Conclusion

In this paper, we describe an enhancement of CiteSeer^X that makes it possible to use the dataset as a test collection for evaluating bibliographic citation recommendation systems. The dataset has certain advantages over other datasets used hitherto. Specifically, it is significantly larger and more heterogeneous. The collection is available for download from anonymised-url. We also provide a set of baseline performance figures for this collection using elementary content-based techniques. We are in the process of ironing out some limitations that this collection has. We hope that the availability of the refined collection will make it easier to reliably and systematically compare different approaches to bibliographic citation recommendation.



```
<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 Fusion function
prevent</contexts>
<clusterid>44856</clusterid>
```

Figure 5: A context for which only one reference is counter-intuitively regarded as relevant

References

Apté, C.; Damerau, F.; and Weiss, S. M. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems* 12.

Beel, J.; Gipp, B.; and Breitinger, C. To appear. Research paper recommender systems – a literature survey. *International Journal on Digital Libraries*.

Caragea, C.; Silvescu, A.; Mitra, P.; and Giles, C. L. 2013. Can't see the forest for the trees?: a citation recommendation system. In *Proc. 13th ACM/IEEE-CS JCDL*. ACM.

Caragea, C.; Wu, J.; Ciobanu, A.; Williams, K.; Fernndez-Ramrez, J.; Chen, H.-H.; Wu, Z.; and Giles, L. 2014. CiteSeer x: A Scholarly Big Dataset. In *Advances in Information Retrieval*. Springer.

Cleverdon, C. 1997. Readings in information retrieval. Morgan Kaufmann Publishers Inc. chapter The Cranfield Tests on Index Language Devices, 47–59.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proc. 3rd ACM conf. Digital libraries*, 89–98. ACM.

Harman, D. 2011. *Information Retrieval Evaluation*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

Hiemstra, D. 2001. *Using language models for information retrieval*. Ph.D. Dissertation, University of Twente.

Huang, W.; Kataria, S.; Caragea, C.; Mitra, P.; Giles, C. L.; and Rokach, L. 2012. Recommending citations: translating papers into references. In *Proc. 21st ACM CIKM*, 1910–1914.

Lewis, D. D. 2004. Reuters-21578 text categorization test collection. http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt.

Lu, Y.; He, J.; Shan, D.; and Yan, H. 2011. Recommending citations with translation model. In *Proc. 20th ACM CIKM*, 2017–2020.

McNee, S. M.; Albert, I.; Cosley, D.; Gopalkrishnan, P.; Lam, S. K.; Rashid, A. M.; Konstan, J. A.; and Riedl, J. 2002. On the recommending of citations for research papers. In *Proc. 2002 ACM conf. Computer supported cooperative work*, 116–125. ACM.

Ponte, J. M., and Croft, W. B. 1998. A language modeling approach to information retrieval. In *Proc. 21st ACM SIGIR*, 275–281.

Porter, M. F. 1997. Readings in information retrieval. chapter An Algorithm for Suffix Stripping.

Ritchie, A.; Robertson, S.; and Teufel, S. 2008. Comparing citation contexts for information retrieval. In *Proc. 17th ACM CIKM*, 213–222.

Ritchie, A.; Teufel, S.; and Robertson, S. 2006. Creating a test collection for citation-based IR experiments. In *Proc. NAACL HLT*, 391–398.

Ritchie, A.; Teufel, S.; and Robertson, S. 2008. Using terms from citations for IR: some first results. In *Advances in Information Retrieval*. Springer. 211–221.

Robertson, S., and Zaragoza, H. 2009. The probabilistic relevance framework: BM25 and beyond. *Information Retrieval* 3(4):333–389.

Strohman, T.; Croft, W. B.; and Jensen, D. 2007. Recommending citations for academic papers. In *Proc. 30th ACM SIGIR*, 705–706.

Sugiyama, K., and Kan, M.-Y. 2010. Scholarly paper recommendation via user's recent research interests. JCDL '10, 29–38.

Sugiyama, K., and Kan, M.-Y. 2013. Exploiting potential citation papers in scholarly paper recommendation. In *Proc. 13th ACM/IEEE-CS JCDL*, 153–162.

Zhai, C., and Lafferty, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. 24th ACM SIGIR*.