Constructing and evaluating automated literature review systems

Jason Portenoy[1] · **Jevin D. West**[1]

Abstract

Automated literature reviews have the potential to accelerate knowledge synthesis and provide new insights. However, a lack of labeled ground-truth data has made it difficult to develop and evaluate these methods. We propose a framework that uses the reference lists from existing review papers as labeled data, which can then be used to train supervised classifiers, allowing for experimentation and testing of models and features at a large scale. We demonstrate our framework by training classifiers using different combinations of citation- and text-based features on 500 review papers. We use the R-Precision scores for the task of reconstructing the review papers' reference lists as a way to evaluate and compare methods. We also extend our method, generating a novel set of articles relevant to the fields of misinformation studies and science communication. We find that our method can identify many of the most relevant papers for a literature review from a large set of candidate papers, and that our framework allows for development and testing of models and features to incrementally improve the results. The models we build are able to identify relevant papers even when starting with a very small set of seed papers. We also find that the methods can be adapted to identify previously undiscovered articles that may be relevant to a given topic.

Keywords Citation networks · Scholarly recommendation · Big scholarly data · Autoreview

Introduction

Conducting a literature review, or survey, is a critical part of research. As the literature continues to grow and as scholars continue to move across disciplines, synthesizing and highlighting existing findings becomes increasingly important. At the same time, it has become increasingly difficult to identify even a slice of the relevant papers for a given topic (Tsafnat et al. 2013). The problem is that this curatorial process does not scale well. It is

✉ Jason Portenoy
jporteno@uw.edu

Jevin D. West
jevinw@uw.edu

1 University of Washington, Seattle, WA 98105, USA

expensive in both time and human effort. The advent of Big Scholarly Data—the availability of data around published research and the techniques and resources to process it—has led to a flurry of activity in finding automated ways to help with this problem (Ammar et al. 2018; Beel et al. 2016; Kanakia et al. 2019; Ronzano and Saggion 2015; Williams et al. 2015; Zitt et al. 2015}.

Many methods have been developed to recommend relevant papers, using features related to textual similarity, keywords, and structural information such as relatedness in a citation network (Beel et al. 2016). However, a common problem in developing and evaluating these methods is a lack of ground truth. We don't know whether our methods are actually selecting relevant papers or topics. This is a general problem in recommender research, but especially so for scholarly papers, given the specialized knowledge needed to evaluate quality and relatedness.

In this paper, we present an approach to this problem that leverages the references in existing review papers as an approximation to ground truth. We assume that references in a review represent at least a subset of papers relevant to a given topic. Using this abundant labeled data within the thousands of reviews in the literature, we are able to frame the collection of a literature survey as a supervised learning problem. Within this supervised framework, we are able to evaluate, at least to some degree, the quality of methods aimed at automatically synthesizing scientific knowledge.

With this framework in place, we demonstrate how supervised learning models can be used to identify relevant papers for review, deriving features from the metadata associated with an article. These features include citations and the groups of papers that can be derived by clustered citation networks (Fortunato 2010). They also include text features derived from the similarity in paper titles. However, any set of related features (authors, disciplines, etc.) could be incorporated.

Using the reference list from a single review article as a benchmark, we develop methods for recapturing those references automatically using the features noted above ("Application to a single review article" section). We then extend this method beyond one review article and apply the methods to a large group of review articles (Sects. "Large-scale study on multiple review papers and Extended analysis" sections). Finally, we apply the methods to identify relevant papers in the fields of science communication and misinformation studies. We invite domain experts to validate our results ("Exploring scientific fields using automated literature review" section). We make code and sample data for this project freely available at https://github.com/h1-the-swan/autoreview.

The main contribution of this work is a novel framework for constructing and evaluating automated methods for generating references for literature surveys at a large scale. This work builds off of a BIRNDL workshop paper presented at SIGIR 2019 (Portenoy and West 2019). We have extended this work in several ways: running thousands of experiments to assess how the methods perform using various review articles, sets of features, and data splits; expanding the background literature review; and reporting results from expert feedback on our exploration of new fields.

Background

There have been several previous attempts at automated or semi-automated literature surveys (Belter 2016; Chen 2012; Janssens and Gwinn 2015; Jha et al. 2013; Silva et al. 2016). These approaches have tended to be smaller scale and rely on more qualitative means of

evaluations, which are difficult to replicate and compare across studies. For example, Chen (2012) developed a system to aid in writing literature reviews, which was evaluated by having first-year graduate students use it to help them write and submit papers for publication. These student-submitted papers had a high acceptance rate, and one student won a best paper award. This evaluation approach, while creative and compelling, does not scale well. Another study by Silva et al. (2016) applied community detection on citation networks to map papers in two different topics and then apply text analytics to generate taxonomies of terms. This approach allowed for detailed analysis of how subtopics are related within a field, but it relied on keyword searches, which can be an insufficient method of identifying all relevant articles (Greenhalgh and Peacock 2005; Horsley et al. 2011; Larsen et al. 2019).

Recent work has explored the use of review articles as a way of testing automated literature review systems. Belter (2016) used a semi-automated technique to retrieve documents for systematic reviews using citations. Janssens and Gwinn (2015) used co-citation and direct citation networks to identify eligible studies for existing biomedical systematic reviews, starting from one or two known articles. These methods have begun to be used in helping to create new systematic reviews (e.g., Albarqouni et al. 2017). Other studies have used active learning approaches to reduce the workload associated with selecting relevant articles for systematic reviews in the domains of medicine and public health (Miwa et al. 2014; Wallace et al. 2010), law (Cormack and Grossman 2014), and software engineering (Yu et al. 2018; Yu and Menzies 2019).

Automatically identifying papers for surveys is similar to recommending papers, more generally. This topic has been extensively studied within and outside big scholarly data. A recent survey paper on research paper recommender systems (Beel et al. 2016) identified more than 200 articles on the topic published since 1998. The survey notes that the majority of approaches use keywords, text snippets, or a single article as input. Our approach, in contrast, starts with a set of seed papers, which are then expanded upon. Our approach also has the distinction of being able to make use of any combination of various features, enabling us to use both textual and network-based features. Some previous work has built recommender systems which combine text and citation information (Gupta and Varma 2017; Kong et al. 2018). These take a different approach, using embeddings to characterize similarity between articles.

The new research in automated methods for literature reviews is the result of people applying newly available data and computational power to a perennial and worsening problem—that of the need for and difficulty of organizing large bodies of research. This need for efficient literature review, and especially systematic review, is strongly felt in medicine, but it is also a need for all areas of science (Bastian et al. 2010; O'Mara-Eves et al. 2015; Tsafnat et al. 2013). In our work, we aim to provide a framework to help with this research by offering a way to develop and test literature review generation and recommendation at a large scale.

Data and methods

Data

The network data used in our analysis come from a recent snapshot of the Web of Science (WoS) citation index consisting of 1,269,262,278 directed citation links between
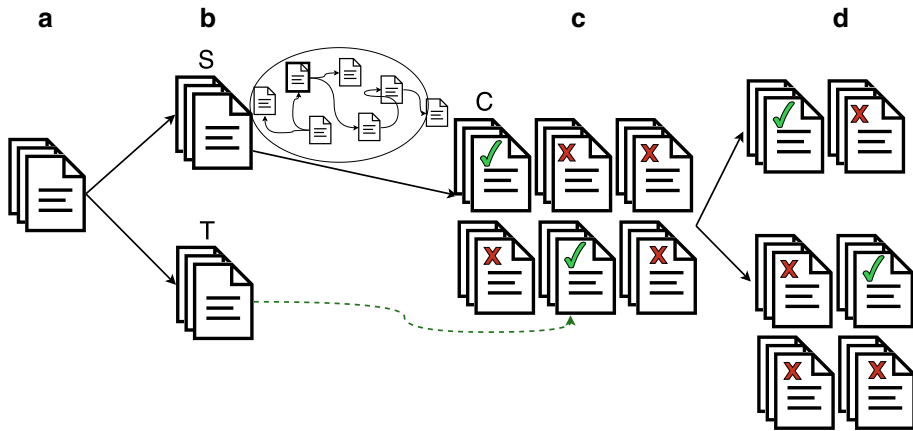
Fig. 1 Schematic of the framework used to collect data for development and testing of a supervised literature review classifier. **a** Start with an initial set of articles (i.e., the bibliography of an existing review article). **b** Split this set into seed papers (S) and target papers (T). **c** Collect a large set of candidate papers (C) from the seed papers by collecting in- and out-citations, two degrees out. Label these papers as positive or negative based on whether they are among the target papers (T). **d** Split the candidate papers into a training set and a test set to build a supervised classifier, with features based on similarity to the seed papers (S)

163,830,918 papers. The data set contains paper-level metadata, such as titles, abstracts, publication dates and venues, and authors. We used WoS because it is one of the most comprehensive bibliographic datasets, covering a large number of articles across most scientific fields. WoS also identifies certain articles as review papers, which was convenient for this project.

We removed some papers from the full data set. In order to reduce the network to a size that we could cluster (see "Features" section), we removed all papers that had no outgoing citations, and any paper that was only cited once (many of these actually appeared to be placeholder data, for which WoS could not fully identify the cited paper). We also removed papers which were missing all metadata, such as publication year and title. This cleaned data set had 55,271,946 papers, and 1,020,164,414 directed citation links.

Identifying candidate papers and setting up the supervised learning problem

Our procedure is presented in Fig. 1. The first step is to randomly split the papers into a set of "seed" papers and a set of "target" papers. We are imagining a researcher who is starting with a set of papers relating to a topic (the seed papers). This researcher wants to expand this set to find the other relevant and important papers in the topic. The target papers can be thought of as the set of papers the researcher has not yet included. Ideally, we would like to search for these target papers within the total set of papers in our data set. However, it is infeasible to generate features and train models using the total set of 55 million papers. To narrow the total set to a more reasonable number of candidate papers, we collect all of the papers that have either cited or been cited by the seed papers. We then go one more degree out, taking all of the papers that have cited or been cited by all of those. We follow a second degree of citations because following direct citations is often not sufficient to identify all relevant literature (Janssens and Gwinn 2015; Robinson et al. 2014). This process of

following in- and out-citations imitates the recommended practice for a researcher looking for papers to include in a survey, but at a larger scale (Webster and Watson 2002). The resulting set of papers, while large (generally around 500K to 2M), is manageable enough to work with. We have found that this method, using different samples for the seed papers, reliably generates sets of papers that contain all or nearly all of the target papers (see "Results" section and Table 2). We label each candidate paper positive or negative depending on whether it is one of the target papers. The goal is to identify the positive (target) papers among the many candidate papers. At this point, we split the candidate papers into training and test sets in order to build classifiers.

Features

Our next step is to generate features to use in a classification model. One feature we use involves the use of citation communities. The structure of the citation network, in which nodes represent papers and edges are directed citations between them, contains patterns about the relatedness of papers. Citation communities are groups of papers that tend to have more connections within-community than outside it. To extract these communities, we used Infomap to cluster the citation network (Rosvall and Bergstrom 2008). Infomap is a well known unsupervised community detection algorithm based on principles of information compression. Identifying clusters in a network of tens of millions of documents is computationally expensive, so we developed a two-step approach to cluster the full network.[1] In the first clustering step, we identified a non-hierarchical clustering of the full network using a parallelized version of Infomap (Bae et al. 2013). This process took 5.3 hours on a machine with 32 cores. 5,513,812 clusters were identified in this way. In the second step, we further processed these clusters to identify hierarchical structure, which is something the parallelized version of Infomap cannot do. We wanted to identify this hierarchy because the structure of science tends to be hierarchical, with smaller communities nested within broader ones. To do this, we used Infomap combined with Apache Spark to further cluster all of the top-level clusters with at least 100 nodes into multi-level, non-overlapping clusters. This second step took about 30 minutes on the same machine. The final clustering had 9,390,263 bottom-level clusters, with a maximum depth of 11, and an average depth of 2.9 (std 0.77).[2]

To incorporate the citation clustering information into classification models, we calculate the average cluster distance between a paper and the seed papers. Distance for two papers i and j is defined as

$$\frac{D_i + D_j - 2D_{LCA}}{D_i + D_j} \tag{1}$$

where D_i and D_j represent the depth in the clustering tree hierarchy of i and j, and D_{LCA} represents the depth of the lowest common ancestor of the two papers' clusters (Djidjev

---

[1] For the clustering, we used the cleaned version of the Web of Science network as described in "Data" section. We used the network after cleaning for citations, but before removing papers with other missing metadata. This version of the network had 73,725,142 nodes and 1,164,650,021 edges.

[2] Since every node is in exactly one cluster (even if the cluster is only one node), and the leaves of the hierarchy tree represent the nodes themselves, the minimum depth in the hierarchy is 2. In this case, the first level is the cluster the node belongs to, and the second level is the node.

et al. 1991).[3] The feature for paper *i* is the average distance to each of the seed papers. We also use PageRank as a measure of citation-based importance (Page et al. 1999).

In addition to these network-based features, we add in a simple text-based feature: the average cosine-similarity of the TF-IDF vector of the paper title to those of the seed paper titles. The purpose of including this feature is to demonstrate how structural- (network) and content- (text) based features can be combined in one model, and can be compared to models with only one or the other. In "Extended analysis" section we extend our analysis to include GloVe word embeddings, and add publication year as an additional feature of paper metadata. There are many other options for features that could be used, including those related to citation or coauthorship patterns, paper text, venue of publication, or any other paper feature that could help identify similarity to the seed papers. Future work will continue this approach, exploring all of these features and how they affect the models' ability to reconstruct the review papers' reference lists. Code and sample data for specifying the features used here are available at https://github.com/h1-the-swan/autoreview.

Results

Application to a single review article

To illustrate how the autoreview process works on a single review article, we use a review article on community detection in graphs (Fortunato 2010). We chose this paper because we are familiar with the topic and could therefore inspect the plausibility of the results. The paper represents a comprehensive review of the topic up to the year of publication (2010). This paper has 262 linked references in our data. We apply the autoreview method using a seed set of 50 papers, randomly sampled from the references. This set of "seed papers" can be thought of as the small set of papers that our imagined researcher above starts with. The remaining 212 papers are "target" papers that we would like to identify.

Table 1 shows the results from five splits, each using a different random seed. The "random seed" is an integer that the sampler uses as a starting point; each different random seed leads to a different split of seed and target sets. Running the process multiple times allowed us to see how the whole system varied when the initial seed papers changed but the review article remained the same. We stopped after five times because generating candidate sets and training models is computationally expensive on the large candidate paper sets. We also wanted to focus our efforts on learning how the system would perform with other review articles ("Large-scale study on multiple review papers" section).

For each run, we split the 262 papers into a set of 50 seed papers and 212 target papers. After collecting candidate papers, we cleaned the data by removing the seed papers, papers for which we did not have titles, and papers published after the year the review paper was published (2010). Each seed (i.e., each row of Table 1) represents one instance of the process in Fig. 1. We report the number of candidate papers in the final set for each run. These sets of candidate papers range in size from 500K to 1.4M papers. In each case, only

---

[3] We divide the standard measure of distance between nodes in a tree by the sum of the nodes' depth. This is because, in the case of hierarchical Infomap clustering, the total depth varies throughout the tree, and the actual depth of the nodes is arbitrary when describing the distance between the nodes. For example, a pair of nodes in the same bottom-level cluster at a depth of level 5 are no closer together than a pair of nodes in the same bottom-level cluster at level 2.

Table 1 Results for autoreview performed on a single review paper, for five different initial random splits of the references into 50 seed papers and 212 target papers

Seed	Num		Network Features			Network + Text		
	Candidates	Recall	Prec at 10/100/1000	R-Prec	Av Prec	Prec at 10/100/1000	R-Prec	Av Prec
1	633,271	0.986	0.8/0.48/0.13	0.425	0.27	0.9/0.77/0.15	0.509	0.472
2	522,098	0.981	0.5/0.4/0.13	0.415	0.227	1/0.72/0.14	0.505	0.462
3	828,817	0.981	0.8/0.45/0.11	0.387	0.23	1/0.82/0.12	0.5	0.429
4	521,479	0.986	0.7/0.42/0.12	0.415	0.245	0.9/0.76/0.13	0.5	0.438
5	1,405,034	0.981	0.8/0.47/0.11	0.396	0.256	1/0.75/0.14	0.491	0.474
Avg	782,140	0.983	0.72/0.44/0.12	0.408	0.245	0.96/0.76/0.14	0.501	0.455

A random forest classifier was trained for each of these splits, for different sets of features. The results shown are for network-based features (average cluster distance and PageRank), and network features + text features (cosine similarity of TF-IDF vectors of paper titles to those of the seed papers)

(at most) 212 of these papers are in the positive class. This parallels the experience of a researcher trying to do an effective survey of a topic—the goal is to find the right papers in a large body of literature that can feel overwhelming. With respect to these candidate sets, this method achieves very high recall: 98.3% on average (standard deviation 0.00258).

After identifying seed papers, target papers, and candidate papers, we generated features for each candidate paper, and labeled them according to whether or not they were among the targets. We then split the candidate papers into a training and test set, and trained classifiers to try to identify the targets. We experimented with a variety of classifiers: logistic regression, SVC (support vector classifier), SGD classifier (stochastic gradient descent), gaussian naive bayes, random forest, and AdaBoost classifier (Murphy 2008). Many of these proved to give poor performance and/or run too slowly, so we proceeded with only random forest, logistic regression, and AdaBoost, selecting the best-performing model for each data set.[4]

Table 1 reports evaluation measures for each of these five models, as well as their averages. These include the *Precision at 10, 100,* and *1000*, the *R-Precision*, and the *Average Precision*. The R-Precision is the fraction of target papers found in the top N papers, where N is the total number of target papers—212 in this case (Manning et al. 2008). The Average Precision is the sum of the precision at k for each rank k of a correctly identified target paper, divided by the total number of target papers. Since the fundamental task is to reconstruct the reference list from the original review paper, we focus our analysis mostly on the R-Precision scores, which characterize exactly how well the models were able to do this (see "Large-scale study on multiple review papers" section for more discussion on the evaluation measures).

Using two network-based features—the average distance between a paper's cluster and those of the seed papers, and the absolute difference of a paper's PageRank score to that of the average seed paper[5]—a random forest classifier identified, on average, 86 of the target

---

[4] Machine learning experiments were conducted using scikit-learn version 0.20.3 running on Python 3.6.9.

[5] Although we only performed ranking and clustering once, it would be ideal to remove all nodes and links past the year of the review paper, as well as the review paper itself, and cluster this network. However, performing a separate clustering for each review paper would be computationally infeasible. Nevertheless, any bias introduced by this should be small, as the clustering method we use considers the overall flow of

Table 2 Summary statistics for the 500 review articles, including the number of references per review (i.e., the seed papers + the target papers to predict), the publication year, the number of candidates generated per initial split of the data, the overall recall for the candidate sets, and precision measures for two sets of features—the network features only (cluster distance and PageRank), and network features + TF-IDF similarity of titles

	Min	Max	Mean	Std	Median
Number of references	200	249	222	14.4	220
Publication year	1939	2016	2001	15.1	2007
Number of candidates	4476	2,152,834	489,418	348,124	453,386
Recall	0.578	1	0.976	0.0469	0.994
Network features					
Precision at 10	0	1	0.355	0.248	0.3
Precision at 100	0	0.69	0.199	0.114	0.18
Precision at 1000	0.002	0.16	0.062	0.0294	0.057
Precision at 10000	0.0016	0.0331	0.0109	0.00307	0.011
R-Precision	0.00625	0.635	0.17	0.0985	0.146
Average precision	0.000734	0.522	0.0891	0.0735	0.0676
Network + Text features					
Precision at 10	0.2	1	0.827	0.147	0.9
Precision at 100	0.06	0.94	0.506	0.129	0.51
Precision at 1000	0.013	0.173	0.0971	0.0256	0.097
Precision at 10000	0.004	0.0331	0.0129	0.00248	0.013
R-Precision	0.0437	0.792	0.385	0.105	0.384
Average precision	0.00713	0.813	0.306	0.12	0.298

papers (40.8%). We also ran the same experiments using the text-based feature of average paper title TF-IDF similarity to the seed paper titles in addition to the network-based features (see "Features" section). This improved the results: the random forest classifiers then identified, on average, 106 of the target papers (50.1%). In the "Appendix", we include some examples of papers ranked by the classifier.

Large-scale study on multiple review papers

We now apply these same methods to multiple review papers. The Web of Science, as part of its citation index data, identifies a subset of articles as review papers (~ 1.4 million papers). We run autoreview on a sample of these reviews to explore how the method performs on a variety of review articles, under varying conditions. We first take a random sample of 500 review articles with between 200 and 250 references. We chose this sample as a starting point in order to hold the number of references relatively constant. We believe that review papers of this size represents the type of review for which this method would be

---

Footnote 5 (continued)

information across multiple pathways, which makes it robust to the removal of individual nodes and links in large networks.

useful—a comprehensive review of a well-defined topic. We also explore results on subsets of larger and smaller review articles in "Extended analysis" section.

For each of these 500 review articles, we took the references and split them into seed and target sets, and tried to use features of the seeds to predict the targets. Table 2 shows summary statistics and results for these papers using the same procedure outlined in the previous section ("Application to a single review article" section). The "number of candidates" refers to the number of papers generated from following two degrees of citations in and out from the seed papers (5 random splits of seed/target for each review paper; 2500 candidate sets in total). These candidate sets are highly imbalanced, with the 150-200 target papers hidden among hundreds of thousands or millions of candidates; again, this is meant to mimic the predicament of the researcher searching for relevant papers in an large pool of candidates. The candidate sets have very high recall, generally including all or missing just one or two papers (with a few outliers in which the citation-based method failed to capture many of the target papers).

For each review article, we gathered the cited papers, and trained models for 5 different random seeds, corresponding to 5 different splits of the data into seed and target papers. We fixed the size of the seed set at 50 papers (in the following section, we explore the effect of varying the size of this seed set). We chose the best-performing model for each split—in most cases, this was a random forest classifier; however, in some instances, a logistic regression or AdaBoost classifier outperformed the random forest.[6]

Again, we report the performance of the classifiers as the *Precision* (at 10, 100, 100, and 10,000), the *R-Precision*, and the *Average Precision* (Manning et al. 2008). The overall goal is to reconstruct the list of held-out target papers from the reference set of the original review articles. Within this task, the relative rank of the different predictions is not especially important. Because of this, we focus the rest of our analysis on the R-Precision scores: the number of correctly predicted target papers among the top *N* papers, where *N* is the number of target papers, divided by *N*. However, looking briefly at the precision at *k* scores—the ratio of target papers identified at different ranks—we can see that the models do tend to give good performance in terms of ranking relevant papers relatively higher. For example, the models given network and text features had, on average, eight papers correctly predicted among the top ten, and 50 among the top 100. We also report the average precision over all target papers as an alternate measure of precision for all relevant documents. This measure is highly correlated to R-Precision (pearson's $r = 0.97$ across all models), so we focus on R-Precision from this point on for simplicity's sake.

Figure 2 shows the distribution of R-Precision scores for 2500 classifiers (five classifiers for each of the 500 review articles, each one trained and tested on a different split of the article's references). The figure shows the classifiers that were given both network (cluster and PageRank) and text (TF-IDF-based similarity of titles) features. The average score was 0.385 (standard deviation 0.105); the highest score was 0.792.

Some of the worst performing review articles tended to be year-specific reviews, e.g., "Germanium : Annual survey covering the year 1972". These particular reviews have temporal constraints that the classifiers did not learn well. Publication date was not even among the features available to these classifiers; adding publication year in the set of

---

[6] We chose to report the best-performing model for each experiment, rather than restricting to a single classifier type. This decision did not have a large effect on the results. We chose to be flexible in which classifier to use because there are differences among the different review articles. We will continue to explore the nature of these differences in future work.
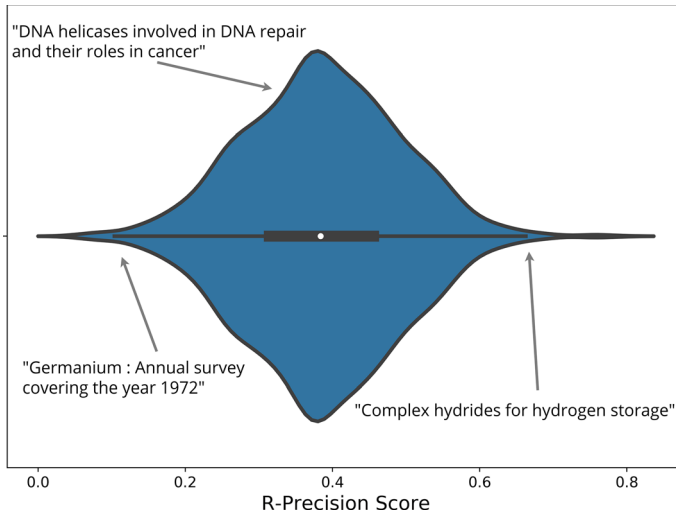
Fig. 2 Violin plot showing the distribution of *R-Precision* scores (number of correctly predicted target papers divided by total number of target papers) for 2500 classifiers, each trained on one of 500 different review articles. The violin plot shows a box plot in the center, surrounded by a mirrored probability distribution for the scores. The distribution is annotated with the titles of three review articles. The review article in the lower tail was one of those which the classifiers did most poorly at predicting references (mean score: 0.14). The one in the upper tail is an example of a review paper whose classifiers performed best (0.65). The one in the middle at the fattest part of the distribution is more or less typical for the review articles in our set (0.39)

features available to the classifiers did cause the performance to improve somewhat. However, in future analysis, these year-specific reviews should be excluded if possible, as they represent a less-typical case with a hard constraint on the potential references. Nevertheless, this type of review article only represented a portion of those in the lower tail, so it is only a partial explanation for the poor performance on these papers. The models tended to perform better with smaller candidate sets (pearson's $r = -0.17$ for the relationship between candidate size and score). This is likely due to the fact that these candidate sets simply had less noise by virtue of them being smaller. However, since the candidates are collected based on random splits of the data, it is not possible to exploit this in order to improve performance (i.e., by limiting the size of the candidate sets).

The analysis to this point has aggregated all review articles together; however, it could be the case that different types of review articles perform differently using these methods. One way to explore this is to look at the discipline of the review articles. Fig. 3 shows the same distribution of R-Precision scores as above, broken down by subject. We used the Web of Science subject labels for the review papers (taking the first one if there were multiple), and aggregated them into broad categories. Most of the reviews analyzed were in Medicine (202), Biology (122), and Natural Sciences (101). Most of the subject groups did not perform significantly differently from each other, suggesting that it is no more difficult to predict the reviews' references in, for example, medicine as it is for those in the natural sciences. Some of the groups on either extreme did show statistically significant differences—e.g., engineering did have higher scores than pyschology/social sciences—but in general the differences between groups were modest at most (pairwise independent t-tests, Bonferroni corrected α of 0.0024). It is
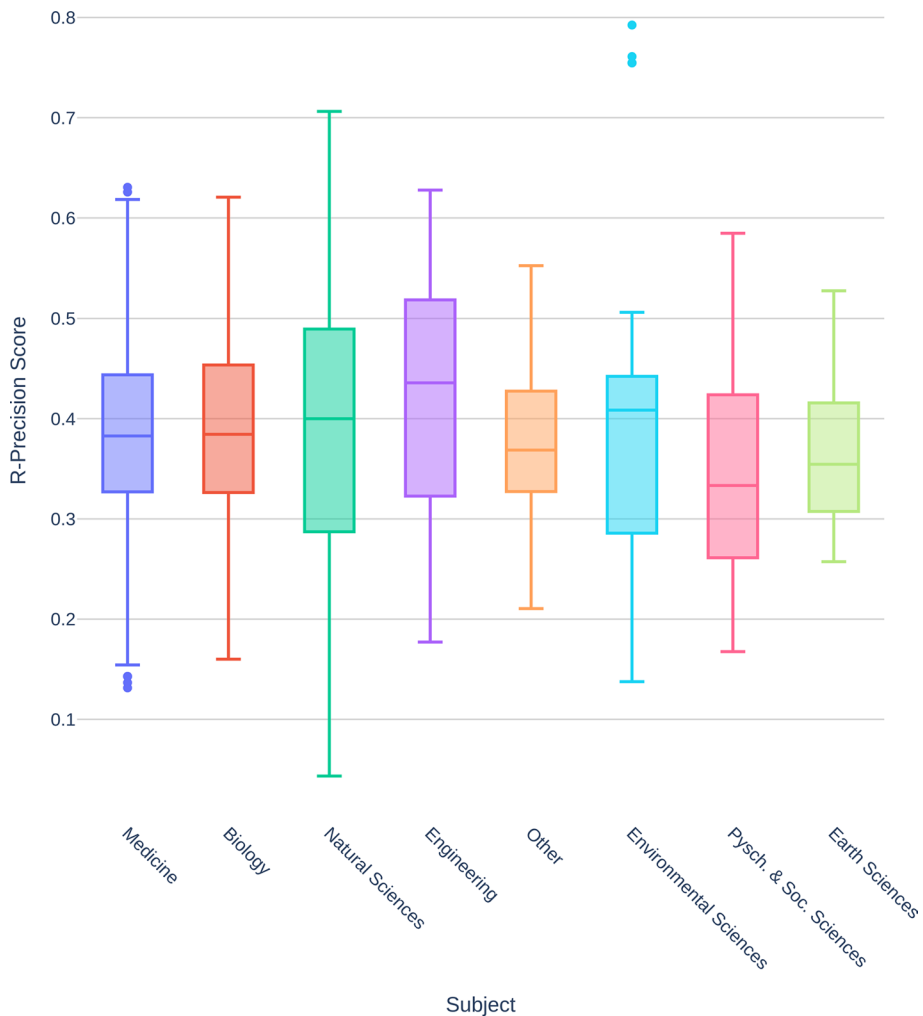
Fig. 3 Box plots of the *R-Precision* scores for the 500 review articles by subject. 50 seed papers, network and TF-IDF title features. See text for discussion

interesting that we did not find any major differences between fields, given that in bibliometric research, findings often do not generalize across different fields.

Extended analysis

We now extend our analysis to explore how the methods perform under various conditions. The three categories of conditions we experiment with are the number of seed/target papers in the initial split of the review references, the features used by the models, and the number of references in the review papers.

Using the same sample of review papers as in the previous section ("Large-scale study on multiple review papers" section), we begin by varying the first two of these: the number
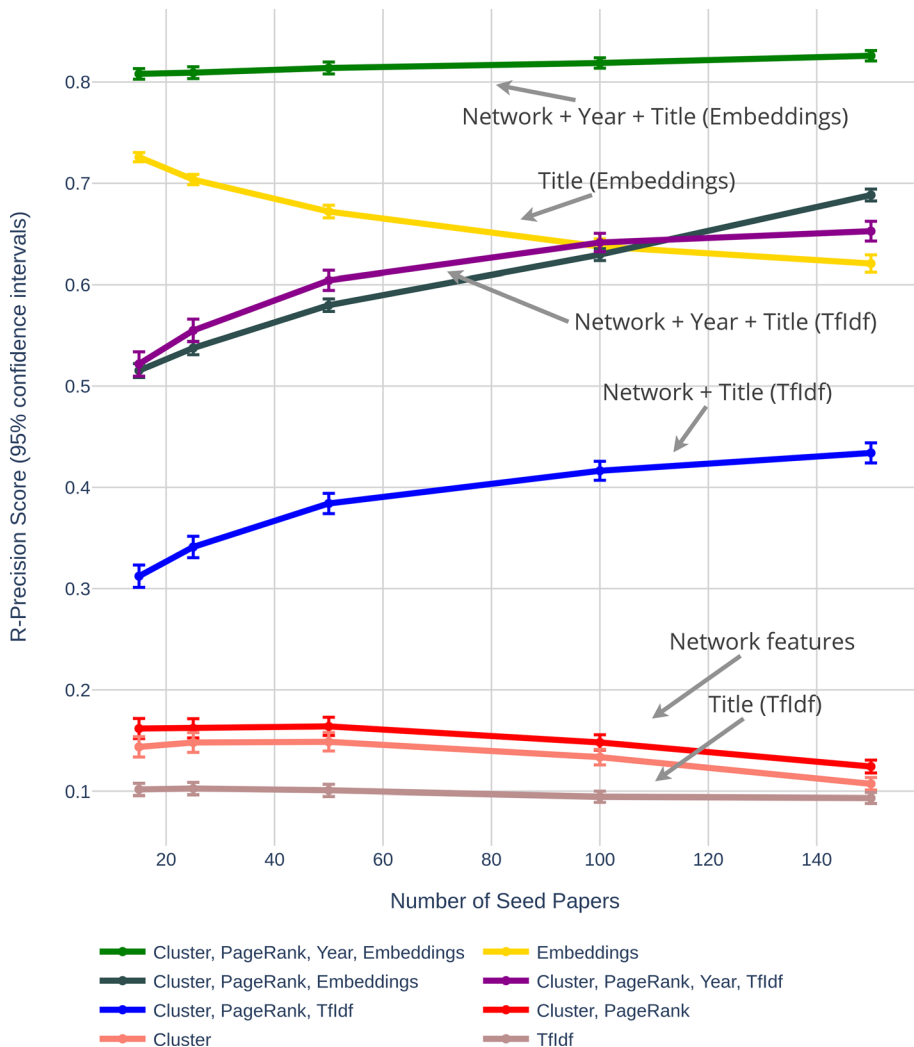
Fig. 4 R-precision scores for autoreview, varying the number of seed/target papers, and the sets of features used. Each point represents the mean of the R-Precision scores for 500 models—5 each for different seed/target splits of the references of 100 review papers. The error bars represent 95% confidence intervals

of seed papers, and the sets of features. We limit our analysis here to a subset of 100 of the previously used 500 articles. This was more computationally tractable, as each combination of seed size and feature set involve training models for five seed/target splits. Fig. 4 shows the R-Precision scores for 8 different sets of features and 5 different numbers of seed papers.

Varying features Each line in Fig. 4 represents the performance of classifiers using different sets of features to rank and identify target papers, with better performing feature sets on top. Using only the TF-IDF information for paper titles gave the worst performance (∼ 0.1). Using network features alone—either cluster information, or cluster information

combined with the paper's PageRank scores—resulted in somewhat higher scores than TF-IDF features (~ 0.15, a 50% improvement over TF-IDF). Combining network and text features, as we saw in the previous section, gave a large boost in performance, with scores around 0.4. Adding another feature from the paper metadata—the publication year—gave another boost, with scores around 0.6.[7] We believe that this improvement is because topics in science tend to be situated in a given period in time. By giving the model information about the publication years of papers, it is better able to identify the important papers in the field.

In order to test more sophisticated text features, we also explored models using title embeddings. For each paper title, we found the average word vector from 300-dimensional GloVe embeddings.[8] We used as a feature the cosine similarity between this vector and the mean of the title vectors for the seed papers. These features tended to perform very well; in fact, using embeddings alone absent any other features tended to give scores higher than most other sets of features that did not include embeddings. The best performing models we tested were ones that combined all types of features—word embeddings, network features, and publication year. These models had R-Precision scores around 0.81.[9]

Varying seed size Each point along the x-axis of Fig. 4 represents results from starting with different sizes of seed/target splits. For example, for each leftmost point, the autoreview process began for each of the 100 review papers by randomly splitting the 200-250 references into a seed set of 15 seed papers and 185-235 target papers, with the target papers then used to generate the candidate sets.Again, this procedure was done with five different random seeds for each review, for each seed size (15, 25, 50, 100, and 150).[10]

Intuitively, we might expect performance to increase along with the size of the seed set, since with more seeds, the classifiers have more knowledge of how similar papers should look, and fewer target papers to predict. We do see this trend for some of the feature sets—for example, with network + TF-IDF, and network + TF-IDF + publication year. Notably, for each of these, the scores for the smaller seed sets are only modestly lower than the largest seed sets, which suggests that this method can perform fairly well even with only a handful of seed papers. On the other hand, some of the feature sets do not improve with more seed papers. The classifiers using title embeddings alone is the most extreme of these: these models actually perform best with the fewest number of seed papers, and performance decreases as the number of seed papers increases. While the reason for this is not entirely clear, it may due to a tradeoff between having more seed papers—which means more information for the classifier to use—but fewer target papers—which means the classifier has to identify the target papers higher up in the rankings in order to get a high score.

Varying size of review papers Figure 5 shows the average R-Precision scores when starting with review articles with reference papers of varying length. The medium size

---

[7] The actual feature used was the absolute difference between a paper's publication year and the mean publication year of the seed papers.

[8] We used the spaCy library (version 2.2.3) with a pretrained English language model (`core_web_lg` version 2.2.5).

[9] The models that had both network and title embedding features, but not publication year ("Cluster, PageRank, Embeddings"), performed worse in general than models with embeddings alone, with scores tending to be between 0.5 and 0.7. The reason for this is unclear.

[10] Since the same random seeds (1, 2, 3, 4, 5) were used each time, the smaller seed sets are always subsets of the larger ones. For example, for a given review article and a given random seed, the 100 seed papers identified are all included in the set of 150; the set of 50 seed papers are all included in both the set of 100 and 150; and so on.
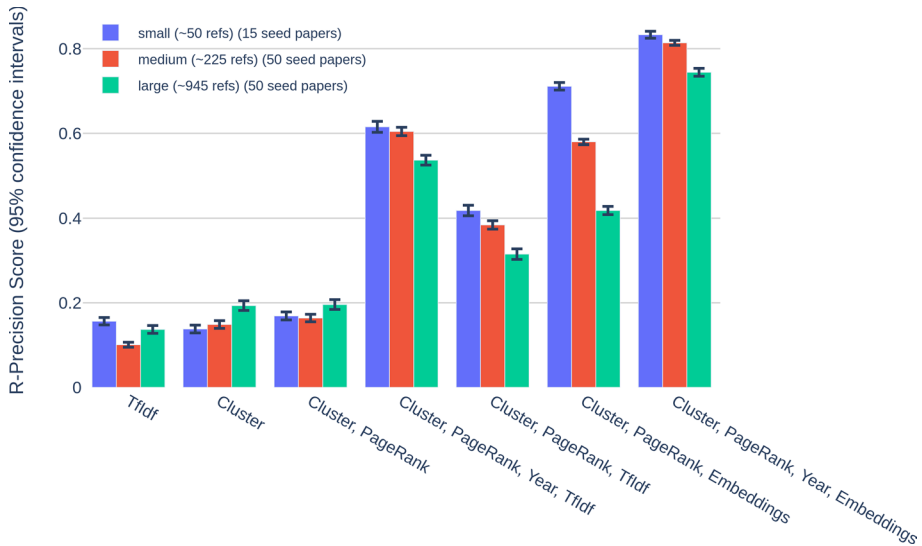
Fig. 5 Average R-Precision scores for different size review articles. The middle (red) bar for each feature set represents the average score for the same 100 review articles using the same procedure as in Fig. 4 (seed size 50). The other two bars in each group represent a different set of review articles, the left a set of 100 smaller reviews (50 references on average), the right a set of 100 larger reviews (945 references on average). Error bars represent 95% confidence intervals. (Color figure online)

articles are the same 100 as above, with a seed size of 50. The small review articles are a different set of reviews that have an average of 50 references, with 15 of these references used as the seed papers. The large reviews are another set of reviews that have on average 945 references, with a seed size of 50. These results are largely consistent with those above. Models with only network or only TF-IDF features all perform about the same, regardless of review paper size. For other feature sets, small review papers tend to perform better than larger ones, but this may be a function of the ratio of seed papers to target papers (as seen in many of the models in Fig. 4), and not due to any inherent differences between these groups of review papers.

Exploring scientific fields using automated literature review

The method we introduce can be adapted as a tool for exploring key papers in an emerging field. In this use case, it is the papers the classifier "misses" that we are interested in. The classifier, attempting to predict the target papers, assigns a confidence score to each of the candidate papers. We are interested in those candidate papers which received a high score, yet were not actually target papers. In the classic classification task, these would be considered misidentified, but in this task we consider the possibility that their similarity to the seed papers may make them relevant papers for this field. This is consistent with Belter's suggestion of "supplement[ing] the traditional method by identifying relevant publications not retrieved through traditional search techniques" (Belter 2016). As a case study, we applied this method to papers in the emerging field of misinformation studies, which pulls research from psychology, risk assessment, science communication, computer science, and others.

As part of this case study and in collaboration with the National Academy of Sciences (NAS), we curated a collection of important papers in this field[11] and used this collection as a seed set to identify other related papers that might have been missed by our more manual methods. Evaluating these results brings us back to shaky territory where we do not have ground truth. However, conversations with domain experts interested in formally characterizing these fields have been encouraging, suggesting the utility of these methods in identifying relevant papers. The original seed papers and the extended bibliography of machine-identified and ranked papers can be found at http://www.misinformationresearch.org.

Leveraging the expertise of the NAS scientists, we are also studying how well these methods can identify papers in a somewhat more established field. We used a seed set of curated papers in the field of Science Communication to identify and rank additional papers. The seed set consisted of 274 papers collected from a 2017 National Academies report on science communication (National Academies of Sciences 2017). We performed five different splits of these papers into seed and target sets (see Fig. 1). For each of these, we generated large sets of candidate papers from Web of Science, and then trained random forest models to rank candidates based on the citation- and title-based features described in "Application to a single review article" section. For each candidate paper, we aggregated the results of the five classifiers by taking the sum of the models' predicted probabilities. We then provided the evaluators with a list of the top 1,000 papers for evaluation that were not in the original seed set. Three domain experts have evaluated this data set, one independently, the other two working together. They made binary relevance judgments for each of the 1,000 papers, with the instructions: "identify any references that the algorithm picked up that don't belong in the field of science communication." The first rater judged 947 (95%) of the references to be relevant, while the other two judged 872 (87%) to be relevant (moderate inter-rater reliability between the two ratings: Cohen's $\kappa = 0.37$). We plan to make use of expert evaluations to assess how useful this approach could be in other fields, including misinformation studies.

Discussion

Our results suggest that it is possible to use automated methods to identify many of the most relevant papers for a literature review, starting from a large set of candidate papers. We believe that, by trying new features and tuning model parameters, we can increase performance and learn more about what distinguishes these papers. We have also seen promise in using these methods to build novel surveys of topics from a set of seed papers. An important area of future work will be collecting more expert-labeled evaluations to validate and improve this approach.

Running these experiments on our samples of review articles required thousands of hours of computation on a supercomputing cluster. However, applying the methods to a single set of references (as in "Exploring scientific fields using automated literature review" section), is much less intensive, and does not necessarily require these resources that may not be broadly accessible or scalable to a general audience.

---

[11] See Data and Methods at http://www.misinformationresearch.org for details

Previous work on automated methods for literature review have tended to use a small number of hand-selected systematic review articles (Belter 2016), or a small number of scientific fields (Silva et al. 2016). The small scale and close qualitative approach can provide a lot of insight, but makes it hard to specify benchmarks to generalize and compare different methods. Our experimental approach, on the other hand, gathers many review papers and applies general techniques, allowing for a much larger pool of labeled data.

We found that we were able to identify many of the references of review articles in a variety of research areas. Our methods also missed many references, ranking other articles more highly than the ones in the original reference list. However, it seems that these "incorrect" articles may actually have value: they may be relevant articles that were missed by the review papers' authors. We found some support for this with the help of domain experts, who found that many of the "misclassified" articles were in fact relevant to the given field. While the precision scores attainable by these methods represent a good goal when making improvements, it is worth noting that in many cases, the failures of the classifiers may actually indicate valuable papers that have been overlooked.

Furthermore, we see potential in using this framework to develop and evaluate methods for literature survey generation and related problems such as scholarly recommendation and field identification. The objective we propose for our modeling task—accurately finding all of the remaining references from a review paper given a held out sample of seed papers from those references—is not a perfect one. We assume that the references in a review paper represent domain experts' best attempt to collect the relevant literature in a single research topic; however, there exist several different types of review article (systematic review, meta-analysis, broad literature survey, etc.), and our current method ignores potential nuance between them. Additionally, we assume that every article in a review paper's bibliography is a relevant article to be included in a field's survey; in practice, an article can be cited for many different reasons, even within a review article. Despite these limitations, the large amount of available data allows our framework to provide a means of experimenting with and developing methods for automated literature surveys. There are many review articles similar to the ones we used that have their bibliographies available and so it will be possible to do this development and analysis on a large scale across many domains. Using this framework, it will be possible to empirically evaluate novel features for their use in identifying papers relevant to a survey in a given topic.

Appendix

Example of autoreview results

Below is a sample of results (random samples of true positives, false positives, true negatives, and false negatives) from the autoreview classifier using the references from Fortunato (2010)—a review on Community Detection in Graphs—with a random seed of 5. The "Rank" represents the position of the candidate paper when ordered descending by the classifier's score. The false positives, while not in the original reference list, still seem

to be relevant to the topic (e.g., "Overlapping Community Search for Social Networks").
The true negatives tend to have lower scores than the false negatives, suggesting that the
assigned score does tend to predict relevant documents, even if they are below the cutoff.

True Positives

Rank	Title	Year
10	Role Models For Complex Networks	2007
21	Adaptive Clustering Algorithm For Community Detection In Complex Networks	2008
24	Random Field Ising Model And Community Structure In Complex Networks	2006
50	Bayesian Approach To Network Modularity	2008
58	The Effect Of Size Heterogeneity On Community Identification In Complex Networks	2006
76	Loops And Multiple Edges In Modularity Maximization Of Networks	2010
97	Synchronization Interfaces And Overlapping Communities In Complex Networks	2008
118	The Analysis And Dissimilarity Comparison Of Community Structure	2006
119	Searching For Communities In Bipartite Networks	2008
208	Epidemic Spreading In Scale-Free Networks	2001

False Positives

Rank	Title	Year
72	Modularity From Fluctuations In Random Graphs And Complex Networks	2004
94	Clustering Coefficient And Community Structure Of Bipartite Networks	2008
98	Detecting Overlapping Community Structures In Networks	2009
106	Size Reduction Of Complex Networks Preserving Modularity	2007
129	Extracting Weights From Edge Directions To Find Communities In Directed Networks	2010
146	Identifying The Role That Animals Play In Their Social Networks	2004
150	Seeding The Kernels In Graphs: Toward Multi-Resolution Community Analysis	2009
159	Overlapping Community Search For Social Networks	2010
162	Modularity Clustering Is Force-Directed Layout	2009
185	Cartography Of Complex Networks: Modules And Universal Roles	2005

True Negatives

Rank	Title	Year
2967	Graph Models Of Complex Information-Sources	1979
120959	Parallel Distributed Network Characteristics Of The Dsct	1992
322251	Hidden Semantic Concept Discovery In Region Based Image Retrieval	2004
327308	A Multilevel Matrix Decomposition Algorithm For Analyzing Scattering From Large Structures	1996
394850	Multiple-Model Approach To Finite Memory Adaptive Filtering	1992
749175	Statistical Computer-Aided Design For Microwave Circuits	1996
943999	Segmental Anhidrosis In The Spinal Dermatomes In Sjogrens Syndrome-Associated Neuropathy	1993
1121787	Rheological And Dielectric Characterization Of Melt Mixed Polycarbonate-Multiwalled Carbon Nanotube Composites	2004
1177851	Explaining The Rate Spread On Corporate Bonds	2001
1256866	The Cyanobacterial Cell Division Factor Ftn6 Contains An N-Terminal Dnad-Like Domain	2009

False Negatives

Rank	Title	Year
259	Heterogeneity In Oscillator Networks: Are Smaller Worlds Easier To Synchronize?	2003
324	Assessing The Relevance Of Node Features For Network Structure	2009
385	The Use Of Edge-Betweenness Clustering To Investigate Biological Function In Protein Interaction Networks	2005
6605	A Measure Of Betweenness Centrality Based On Random Walks	2005
19863	On Decomposition Of Networks In Minimally Interconnected Subnetworks	1969
59900	Objective Criteria For Evaluation Of Clustering Methods	1971
139178	Optimization With Extremal Dynamics	2001
250583	The Tie Effect On Information Dissemination: The Spread Of A Commercial Rumor In Hong Kong	2002
281952	Compartments Revealed In Food-Web Structure	2003
1203248	Dynamic Asset Trees And Portfolio Analysis	2002

References

Albarqouni, L., Doust, J., & Glasziou, P. (2017). Patient preferences for cardiovascular preventive medication: A systematic review. *Heart*, *103*(20), 1578–1586. https://doi.org/10.1136/heartjnl-2017-311244.

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., & Downey, D., et al. 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. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-3011. https://www.aclweb.org/anthology/N18-3011

Bae, S. H., Halperin, D., West, J., Rosvall, M., Howe, B. (2013). Scalable flow-based community detection for large-scale network analysis. In *2013 IEEE 13th international conference on data mining workshops* (pp. 303–310). https://doi.org/10.1109/ICDMW.2013.138

Bastian, H., Glasziou, P., & Chalmers, I. (2010). Seventy-five trials and eleven systematic reviews a day: How will we ever keep up? *PLOS Medicine*, *7*(9), e1000326. https://doi.org/10.1371/journal.pmed.1000326.

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, *17*(4), 305–338. https://doi.org/10.1007/s00799-015-0156-0.

Belter, C. W. (2016). Citation analysis as a literature search method for systematic reviews. *Journal of the Association for Information Science and Technology*, *67*(11), 2766–2777. https://doi.org/10.1002/asi.23605.

Chen, T. T. (2012). The development and empirical study of a literature review aiding system. *Scientometrics*, *92*(1), 105–116. https://doi.org/10.1007/s11192-012-0728-3.

Cormack, G. V., Grossman, M. R. (2014). Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In: Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval, SIGIR '14, pp. 153–162. ACM, New York, NY, USA. https://doi.org/10.1145/2600428.2609601. http://doi.acm.org/10.1145/2600428.2609601. Event-place: Gold Coast, Queensland, Australia

Djidjev, H. N., Pantziou, G. E., & Zaroliagis, C. D. (1991). Computing shortest paths and distances in planar graphs. In J. L. Albert, B. Monien, & M. R. Artalejo (Eds.), *Automata, languages and programming* (pp. 327–338). Berlin: Springer.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*(3–5), 75–174.

Greenhalgh, T., & Peacock, R. (2005). Effectiveness and efficiency of search methods in systematic reviews of complex evidence: Audit of primary sources. *BMJ*, *331*(7524), 1064–1065. https://doi.org/10.1136/bmj.38636.593461.68.

Gupta, S., Varma, V. (2017). Scientific Article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th international conference on world wide web companion, WWW '17 Companion* (pp. 1267–1268). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland. https://doi.org/10.1145/3041021.3053062.

Horsley, T., Dingwall, O., & Sampson, M. (2011). Checking reference lists to find additional studies for systematic reviews. *Cochrane Database of Systematic Reviews*,. https://doi.org/10.1002/14651858.MR000026.pub2.

Janssens, A. C. J. W., & Gwinn, M. (2015). Novel citation-based search method for scientific literature: Application to meta-analyses. *BMC Medical Research Methodology*, *15*(1), 84. https://doi.org/10.1186/s12874-015-0077-z.

Jha, R., Abu-Jbara, A., Radev, D. (2013). A system for summarizing scientific topics starting from keywords. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: short papers)* (pp. 572–577).

Kanakia, A., Shen, Z., Eide, D., Wang, K. A scalable hybrid research paper recommender system for microsoft academic. In *The World Wide Web conference, WWW '19* (pp. 2893–2899). Association for Computing Machinery. https://doi.org/10.1145/3308558.3313700.

Kong, X., Mao, M., Wang, W., Liu, J., & Xu, B. (2018). VOPRec: Vector representation learning of papers with text information and structural identity for recommendation. *IEEE Transactions on Emerging Topics in Computing*,. https://doi.org/10.1109/TETC.2018.2830698.

Larsen, K. R., Hovorka, D., Dennis, A., & West, J. (2019). Understanding the elephant: The discourse approach to boundary identification and corpus construction for theory review articles. *Journal of the Association for Information Systems*, *20*, 7. https://doi.org/10.17705/1jais.00556.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* (1st ed.). New York: Cambridge University Press.

Miwa, M., Thomas, J., O'Mara-Eves, A., & Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics*, *51*, 242–253. https://doi.org/10.1016/j.jbi.2014.06.005.

Murphy, K. P. (2010). *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press.

National Academies of Sciences. (2017). *Engineering, and Medicine and others: Communicating science effectively: A research agenda*. National Academies Press.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., & Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: A systematic review of current approaches. *Systematic Reviews*, *4*(1), 1–22. https://doi.org/10.1186/2046-4053-4-5.

Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab. http://ilpubs.stanford.edu:8090/422.

Portenoy, J., & West, J. D. (2019). Supervised learning for automated literature review. *BIRNDL*, *2019*, 9.

Robinson, K. A., Dunn, A. G., Tsafnat, G., & Glasziou, P. (2014). Citation networks of related trials are often disconnected: Implications for bidirectional citation searches. *Journal of Clinical Epidemiology*, *67*(7), 793–799. https://doi.org/10.1016/j.jclinepi.2013.11.015.

Ronzano, F., Saggion, H. (2015). Dr. inventor framework: Extracting structured information from scientific publications. In *International conference on discovery science* (pp. 209–220). Springer.

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*(4), 1118–1123.

Silva, F. N., Amancio, D. R., Bardosova, M., Costa, L.D.F., & Oliveira, O.N. (2016). Using network science and text analytics to produce surveys in a scientific topic. *Journal of Informetrics*, *10*(2), 487–502. https://doi.org/10.1016/j.joi.2016.03.008.

Tsafnat, G., Dunn, A., Glasziou, P., & Coiera, E. (2013). The automation of systematic reviews: Would lead to best currently available evidence at the push of a button. *BMJ*, *346*(7891), 8–8.

Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C., & Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, *11*(1), 55. https://doi.org/10.1186/1471-2105-11-55.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii–xxiii.

Williams, K., Wu, J., Choudhury, S. R., Khabsa, M., Giles, C. L. Scholarly big data information extraction and integration in the CiteSeerx digital library. In *2014 IEEE 30th international conference on data engineering workshops* (pp. 68–73). IEEE. https://doi.org/10.1109/ICDEW.2014.6818305. http://ieeexplore.ieee.org/document/6818305/.

Yu, Z., Kraft, N. A., & Menzies, T. (2018). Finding better active learners for faster literature reviews. *Empirical Software Engineering*, *23*(6), 3161–3186. https://doi.org/10.1007/s10664-017-9587-0.

Yu, Z., & Menzies, T. (2019). FAST2: An intelligent assistant for finding relevant papers. *Expert Systems with Applications*, *120*, 57–71. https://doi.org/10.1016/j.eswa.2018.11.021.

Zitt, M. (2015). Meso-level retrieval: IR-bibliometrics interplay and hybrid citation-words methods in scientific fields. *Delineation*, *102*(3), 2223–2245. https://doi.org/10.1007/s11192-014-1482-5.