IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 22, 2021, accepted December 15, 2021, date of publication December 23, 2021, date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3137960

Combining Citation Network Information and Text Similarity for Research Article Recommender Systems

JULIA A. STERLING[1] **AND MATTHEW M. MONTEMORE**[2]
[1]Department of Mathematics, Tulane University, New Orleans, LA 70118, USA
[2]Department of Chemical and Biomolecular Engineering, Tulane University, New Orleans, LA 70118, USA

Corresponding author: Matthew M. Montemore (mmontemore@tulane.edu)

This work was supported in part by the High Performance Computing (HPC) Resources and Services Provided by Technology Services at Tulane University, New Orleans, LA, USA. The work of Julia A. Sterling was supported in part by the Honors Summer Research Program from Tulane University and in part by the Robert and Gayle Longmire Professorship. The work of Matthew M. Montemore was supported by the Louisiana Board of Regents Research Competitiveness Subprogram under Contract LEQSF(2019-22)-RD-A-20.

ABSTRACT Researchers often need to gather a comprehensive set of papers relevant to a focused topic, but this is often difficult and time-consuming using existing search methods. For example, keyword searching suffers from difficulties with synonyms and multiple meanings. While some automated research-paper recommender systems exist, these typically depend on either a researcher's entire library or just a single paper, resulting in either a quite broad or a quite narrow search. With these issues in mind, we built a new research-paper recommender system that utilizes both citation information and textual similarity of abstracts to provide a highly focused set of relevant results. The input to this system is a set of one or more related papers, and our system searches for papers that are closely related to the entire set. This framework helps researchers gather a set of papers that are closely related to a particular topic of interest, and allows control over which cross-section of the literature is located. We show the effectiveness of this recommender system by using it to recreate the references of review papers. We also show its utility as a general similarity metric between scientific articles by performing unsupervised clustering on sets of scientific articles. We release an implementation, ExCiteSearch (bitbucket.org/mmmontemore/excitesearch), to allow researchers to apply this framework to locate relevant scientific articles.

INDEX TERMS Scientific literature, recommender systems, search engines, search methods.

I. INTRODUCTION

When pursuing a line of inquiry, assembling a comprehensive set of relevant knowledge from the scientific literature is a crucial first step. Whether writing a review, trying to answer a question, assessing the feasibility of a research topic, or extracting a dataset from previous studies, there is a clear need to find a comprehensive and focused set of research articles. However, locating this subset of the vast scientific literature can be challenging.

The sheer volume of scientific articles published can be overwhelming, and sifting through it all manually is challenging and often impractical [1]. There are many journals with overlapping scopes, ranging from very specific to very general, and examining even just the publications in a single journal can be time-consuming. Probing through many journals is often prohibitive due to time constraints, causing researchers to turn to automated search methods. Traditional search methods return content that is similar to a handful of keywords, and many of the results may not be relevant. Keyword searching is particularly problematic when dealing with synonyms or multiple meanings of the same terms, which can be quite common in technical contexts.

Recommendations are becoming increasingly important and have been applied in many content domains including music and movies. As data mining and machine learning increase in usage and power, recommendations are taking a progressively more dynamic approach [2].

In addition to recommending scientific articles for researchers to read, the possibility of automated identification of similar articles may be useful for extracting data from the scientific literature. For example, extracting materials

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Luca Bernardi.

science data from the scientific literature can be used to train machine learning models to predict new functional materials [1], [3], [4]. Furthermore, quantifying similarity between scientific articles allows unsupervised machine learning on the literature itself, including clustering [5], [6]. This is related to the concept of science mapping, which aims to visualize or analyze a field of research [7].

Broadly, the approach we take here is to identify scientific articles that are similar to a smaller, input set of articles that are already known to be relevant, somewhat similar to some movie or music recommendation systems that use previous user ratings. Specifically, our approach is analogous to recommending additional songs to add to a given music playlist, rather than recommending songs based on an entire music library or based on a single song. Because researchers may work in multiple subfields, recommendations based on an entire library can be unfocused. On the other hand, recommendations based on a single paper may not be similar to that paper in the intended way.

Here, we describe and demonstrate the efficacy of ExCiteSearch, a research-paper recommender system that utilizes a user-determined combination of citation information and text similarity to find relevant papers. We also share a practical, flexible, and open-source implementation. Because of ExCiteSearch's focus on citations and flexible search methods, ExCiteSearch returns relevant results effectively without overwhelming the user. Having a general measure of distance or similarity between articles is useful for bibliometrics and machine learning, as well as for recommendation, and hence ExCiteSearch can be useful for a variety of applications.

As the volume of scientific literature has increased, research-paper recommender systems have grown in importance due to their effectiveness for handling massive amounts of data and their suitability over keyword-based searching [8]. The most common methodologies employed by various systems are collaborative filtering, content-based filtering, graph networks, and hybrid search methods.

Content-based filtering is one of the most commonly used and researched types of recommendation [9]. Content-based filtering examines the description of an item and compares it to the profile of a user [10]. Creating user profiles and analyzing the description or content of an item consumes a great deal of computing power and also does not take into account the quality or the popularity of an item [11]. Additionally, content-based filtering's reliance on a user profile leads it to only recommend papers that are very similar to the user's knowledge base [12]. Furthermore, the reliance on a user's entire profile can be limiting when the user only wants information on a specific subfield or new interest. Therefore, the use of a user's entire profile is a significant limitation of current content-based filtering approaches.

Alternatively, in collaborative filtering, recommendations are made based on ratings from users. Collaborative systems perform well, but their success is reliant on having a high number of user ratings, so a lack of data can worsen the quality of recommendations [13], [14]. Collaborative filtering works best with complex items such as music and movies where variations in taste cause most of the variation in preferences [15].

Graph-based recommendations are a less common approach, and systems use different criteria, such as citations, author collaborations, or year of publication to determine connections between papers [16]. Citations linking scientific papers are especially useful as a recommendation tool because papers linked by citations have already been identified as being directly related [17]. One difficulty with graph-based citations is the possible lack of citations across multiple communities or subject areas because one academic community may not be aware of related activity in another academic community [18]. Using some measure of content similarity in addition to the citation network may help mitigate this drawback.

There are a number of existing citation analysis tools and techniques that examine citations, but these tools have drawbacks for practical implementation as recommender systems. Many proposed tools and models are only useful for articles available through PubMed or Citeseerx and are therefore are ineffective for a variety of disciplines [35], [36]. There are methods that determine the relevance of a citation by analyzing its position within the text, such as citation intent classification and in-text citations [37] [38], but these rely on parsing the entire text of an article which is often prohibitive due to constraints of time, computer power, and access. Indeed, accurately parsing PDF files programmatically is often difficult. In addition, our work allows a researcher to find different cross sections of the literature (e.g., the same method on different systems vs. different methods on the same system), which is not possible using pre-defined clusters.

All or some of the aforementioned recommendation approaches may be combined into various hybrid techniques. A weighted system computes the score of an item from the results of all recommendation techniques available, whereas a switching hybrid system switches between recommendation classes. A mixed hybrid system presents recommendations from several different recommenders at one time. The use of hybridization can alleviate some of the problems associated with relying only on content-based filtering, collaborative filtering, and other recommendation techniques [15].

ExCiteSearch was designed considering the strengths and weaknesses of these recommendation classes. ExCiteSearch uses a switching hybrid technique, changing between graph-based recommendations and a form of text-based searching depending on the user's preferences. Allowing the user to choose one or both of these recommendation techniques in the search process is intended to overcome the challenges associated with using just one recommendation technique.

II. APPROACH

A. OVERVIEW

We define similarity between publications through a combination of text similarity and connectedness through the

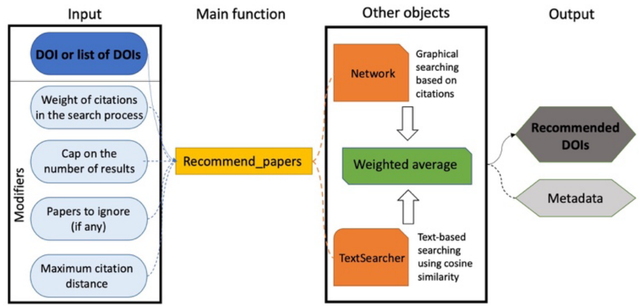
Input
Main function
Other objects
Output
DOI or list of DOIs
Modifiers
Weight of citations in the search process
Cap on the number of results
Papers to ignore (if any)
Maximum citation distance
Recommend_papers
Network
Graphical searching based on citations
Weighted average
TextSearcher
Text-based searching using cosine similarity
Recommended DOIs
Metadata

FIGURE 1. An overview of ExCiteSearch's core architecture. Given an input set of papers and other optional parameters, recommend_papers uses a combination of two underlying functions to identify related papers. Other functions and objects (e.g., thumbs and Grapher) build on this core architecture.

citation network. ExCiteSearch's core functions use this definition of similarity to determine which articles are most relevant to an initial article or set of articles of interest. The primary function for recommending papers, recommend_papers, searches through articles that are connected to any of the articles in the input set through the citation network, quantifying the citation connection by using the number of citations that link articles and then quantifying text similarity by comparing titles and abstracts using cosine similarity. Citations are taken from the reference lists, and citation context is not considered. The objects are the Network object, which represents relationships between articles as a graph, and the TextSearcher object, which handles text-based searching. Figure 1 illustrates some of ExCiteSearch's core functions, objects, and objectives.

B. PRIMARY USER FUNCTIONS

Recommend_papers is ExCiteSearch's main search function, which allows the user to tailor the search process. It can take either a single DOI or a list of DOIs as input and can return either a list of DOIs, a dictionary of DOIs and all their metadata, or a dictionary of DOIs and a subset of their metadata, such as the title and author. Recommend_papers can also put a cap on the number of results returned, which gives the user control over the size of the output.

Recommend_papers enables tailoring of the search process by allowing the user to decide how much weight citation similarity and text similarity will each take in the final results. This works by running TextSearcher's search function on results from Network's search function, and then weighting the results accordingly. Min-max scaling is used on the citation similarity and text similarity to treat the two kinds of similarity on an equal footing. If the citation weight is 0, then only TextSearcher's search function is used, and if the citation weight is 1, then only Network's search function is used. After weighting and combining the two scores, the returned papers are given in order from highest overall score to lowest. Additionally, recommend_papers can also take in a second list of DOIs that it will ignore both by removing them from the citation network and disregarding them when examining text similarity. ExCiteSearch can also read text files containing lists of DOIs as the input to the search or the papers to ignore.

The ''thumbs'' function allows the user to guide the search as it happens and get output from recommend_papers one article at a time. It initially takes in a DOI or list of DOIs. The most relevant article to the input is returned, and the user has the option of opening it in a web browser. The user can then either give the output a ''thumbs up'' or a ''thumbs down'' and the function takes this feedback into account and returns a new paper. This continues until the user decides to quit. This function was inspired by Vannevar Bush's concepts of the memex and associative indexing, where the initial input can be used to automatically select another related publication immediately [19]. This is also similar to certain music and video recommendation systems [20].

Quantify_similarity intakes a pair of DOIs or a list of DOIs and quantifies the similarity between the papers using a user-determined combination of citation data and cosine similarity. Quantify_similarity then either returns a single score, a dictionary of DOIs and scores, or a Pandas DataFrame of relevance scores. This allows quantification of the similarity among a group of papers, which can be useful for machine learning purposes such as clustering.

The Grapher class is a wrapper class that searches with either a single DOI or list of DOIs as its query, using both citation and text similarity, and presents the search results graphically on a scatterplot, with cosine similarity plotted vs. text similarity (Figure 2). Higher cosine similarity and higher citation similarity suggest higher overall similarity. Upon hovering over one of the dots on the scatterplot, the DOI of that result will appear. The results can also be obtained as a dictionary. The articles in Figure 2 with a citation distance of 0.75 are articles that are cited both by the query paper and the papers that it cites.

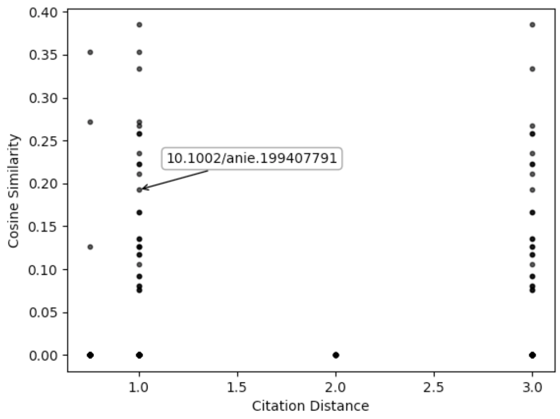

FIGURE 2. An example of Grapher's output for all papers within two citations of an arbitrarily chosen paper [34]. Cosine similarity and citation distance are shown between each paper and the input paper.

C. UNDERLYING OBJECTS

The Network object, which contains all of the citation information, has its own search function that only looks at citation relationships. Citation similarity is quantified by the number of citation connections between an article and the articles in the input set. For example, if an article cites three of the articles in the input set, then the similarity between that article and the input set is three. Citing articles and cited articles are treated the same. The farther away in the citation network a citing article is, the less it contributes to the similarity metric. For example, if a publication is two citations away from an input article, then its connection counts as 1/2, while a publication three citations away counts as 1/3, and so on. This measure of citation similarity is related to the previous concepts of co-citation [21]–[23] and bibliographic coupling [23], [24], and is well-suited as a one-to-many similarity measure. The Network object stores the relationships between papers as a graph using the existing NetworkX package. The Network object also handles its own citation-based search function, which uses NetworkX to determine the closest DOIs in the graph and return them [25].

Additionally, the TextSearcher object, which handles text-based searching, has a search function that calculates similarity by computing the cosine similarity of two publications. Cosine similarity is a continuous, pairwise similarity metric that treats pieces of text as term-frequency vectors, where the similarity between two pieces of text is calculated as the cosine of the angle between the vectors; this is one of the most fundamental techniques in natural language processing and has been shown to be very effective [37]. We also tested term frequency–inverse document frequency and the JSTOR topic inferencer, and found that a simple cosine similarity was the most effective metric. TextSearcher retrieves the abstract, when available, or title, when the abstract is not available, of each publication, removes commonly used words such as "and" and "the" to get text samples X and Y, and then calculates the cosine as $c = |X \cap Y|/(\sqrt{X}\sqrt{Y})$ [26]. TextSearcher's search function uses the cosines of titles or abstracts, when available, to determine similarity.

Network and TextSearcher perform most of ExCiteSearch's main functionalities, but ExCiteSearch is further compartmentalized into several objects that perform its underlying functions. The SciPaper object represents an individual scientific paper. Each SciPaper has a dictionary of attributes attached to it: DOI, title, author, year, journal, abstract, and citations. The SciPaper object also has the function get_info, which can retrieve the metadata of a paper. This function utilizes the OpenCitations [27] and Crossref APIs [28], querying OpenCitations and then Crossref if the paper is not in the OpenCitations database, and can obtain metadata for most of the papers we tested.

In order to increase the speed of ExCiteSearch, the metadata of all these papers is stored in the Corpus object so that a web query does not need to be made every single time an article comes up in the search process. The Corpus object represents a corpus consisting of articles whose metadata has been indexed. The Corpus object also has the capability to add records directly downloaded from Web of Science.

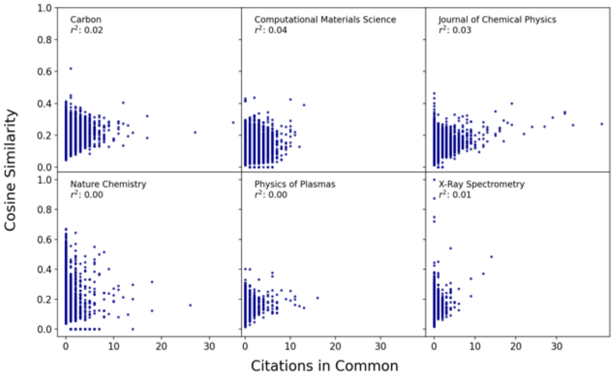

FIGURE 3. The number of citations in common and the cosine similarity between articles from six different journals, on a logarithmic scale. For efficiency, this was done one citation outwards only using the references from each article.

III. RESULTS AND DISCUSSION

To examine the relationship between citation connections and text similarity, we plotted these two quantities against each other for a variety of papers (Figure 3). Specifically, we took 500 papers from *Carbon,* 500 papers from *Journal of Chemical Physics*, 500 papers from *Nature Chemistry,* 500 papers from *Physics of Plasmas,* and 271 papers from *X-Ray Spectrometry*. For simplicity, in this plot we counted the number of papers that a given pair of papers both cite, as a simplification of the citation similarity metric we use in the rest of the work. While it is rare for papers with many citations in common to also have very low cosine similarity, there is very little correlation overall between the number of citations in common and the text similarity. Hence, using both citation similarity and text similarity provides more information than either alone.

There is no well-established benchmark for measuring the efficacy of a research-paper recommender system, so we created a simple, automated test of how well ExCiteSearch may be able to find references for a review paper. Collecting the references for a review paper is often an arduous task, and an improved method for discovering the majority of relevant references could save researchers significant time and effort. To test the utility of ExCiteSearch in this task, we created a function that would attempt to reproduce the reference list of a published review paper (Figure 4). This function was designed to roughly mimic how a human would use ExCiteSearch to find the set of papers needed to write a review. First, a small subset of the references of a given review was chosen randomly. A user would locate these initial references using existing methods, perhaps via a search engine, or would already be aware of them from prior exposure. Then, ExCiteSearch was used to search for papers relevant to this initial set. A user would then add all relevant papers returned to the tentative list of references. To mimic this, we keep any recommended papers that are part of the review's reference list, as these are assumed to be relevant. The new, larger list of references (stored locally in a simple text file) is then

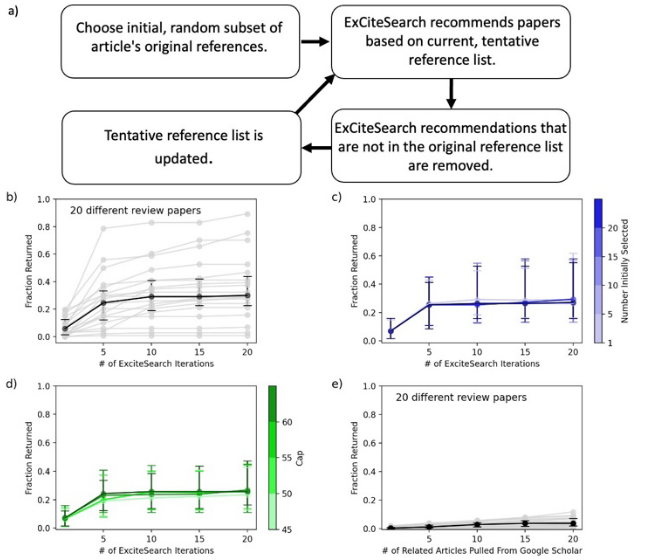

FIGURE 4. a) As a simple test of ExCiteSearch, we randomly chose a subset of papers from a review paper's reference list and used that as an input to recommend_papers. The fraction of the review's reference list that was discovered was tracked as a function of the number of iterations. b) For each review paper, 15 references were initially selected and the cap on the number of results from each iteration was fixed at 60. The results for each reference are shown as a light gray line, while the median and quartiles are shown as the black line and error bars. c) Varying the initial number of references selected, while the cap was fixed at 60. d) Varying the cap on the number of papers returned, with the number of initial references fixed at 15. e) We pulled records from Google Scholar's "related papers" feature for 20 different review papers.

fed iteratively back to ExCiteSearch to find more and more relevant papers.

Using this function, we tested ExCiteSearch's ability to recreate the references of several review papers. We selected ten review papers from Chemical Society Reviews and ten review papers from the Annual Review of Chemical and Biomolecular Engineering [29]–[33], [39]–[53]. ExCiteSearch was given a randomly selected subset of the references of each paper, and the number of results was capped. This was performed multiple times as detailed in Figure 4, with the recommend_papers function removing and ignoring all citation connections through the original review paper. Within the first few iterations, the percentage of the references obtained usually increased quickly (Figure 4b), even when only a small number of references were initially chosen. These results suggest that ExCiteSearch can accurately and efficiently return related papers. A person collecting references to write a review would select an initial set of publications that are representative of the core topic of the review. That is impractical for our test due to lack of domain knowledge for each review, so the initial references used were randomly selected each time. Additionally, a review article may cover more than one subtopic, so a researcher may decide to use ExCiteSearch separately to find recommendations for each subtopic. Therefore, a researcher-guided search would likely be more efficient than our function, which exhibits diminishing returns as the number of iterations increases.

ExCiteSearch was still nearly always effective in finding related papers in just a few iterations when initially fed 15 randomly selected references (Figure 4b). Generally, 3% or less of the reference list was found in one iteration, but in most cases 10-30% was found in five iterations. There is a significant difference between different review papers and between different initial reference sets, due to the random selection of the initial set as well as the variation between reviews, scopes, and the number of references. However, in nearly all cases there is a fast increase in the number of papers found in just a few iterations at some point followed by a leveling off, suggesting that once a certain well-connected set of references is identified, ExCiteSearch can quickly find a significant portion of the references.

Selecting a few more references initially generally has little effect on the rate of discovery of additional papers in the reference list (see Figure 4c). However, there is significant variation based on which papers are initially selected, even when the number selected is held constant. Again, this shows that selecting a highly relevant initial subset can result in very effective searches, and is more important than finding a large initial subset. In all cases, the fraction of references found tends to level off around 5 to 10 iterations. This suggests that ExCiteSearch may identify highly relevant articles early in the search process and the remaining articles may not be as closely related to the core topic. In any case, finding a significant portion of the reference set in just a few iterations can greatly increase the efficiency of assembling the reference list.

Increasing the maximum number of articles returned in each iteration results in a slightly higher fraction of the original reference list being returned for a given number of iterations (Figure 4d). With a higher cap, ExCiteSearch has more information to find relevant papers with, and therefore can do so more quickly. However, because increasing the cap has a small effect, it may be more convenient to use a smaller cap in most cases.

Because most previous recommender systems use different sorts of inputs and/or are not publicly available, it is difficult to directly compare ExCiteSearch to previous systems. Because most researchers rely heavily on search engines to locate relevant literature, we used Google Scholar's "related papers" function to provide a baseline for how difficult it is to gather relevant literature. We pulled records manually for each of the same 20 review papers that were used in Figure 4b (see Figure 4e). Again, a direct comparison is not possible, but it is clear that taking the top 20 results from Google Scholar usually gives only a very small fraction of the reference list, while ExCiteSearch can generally find a significant fraction of the reference list in a few iterations. Indeed, the fraction of the reference list returned from Google Scholar is nearly identical at 15 and 20 papers, showing very fast diminishing returns. Further, Google Scholar can take advantage of the citation connections between the review and the papers it cites, whereas we removed these connections as part of our test.

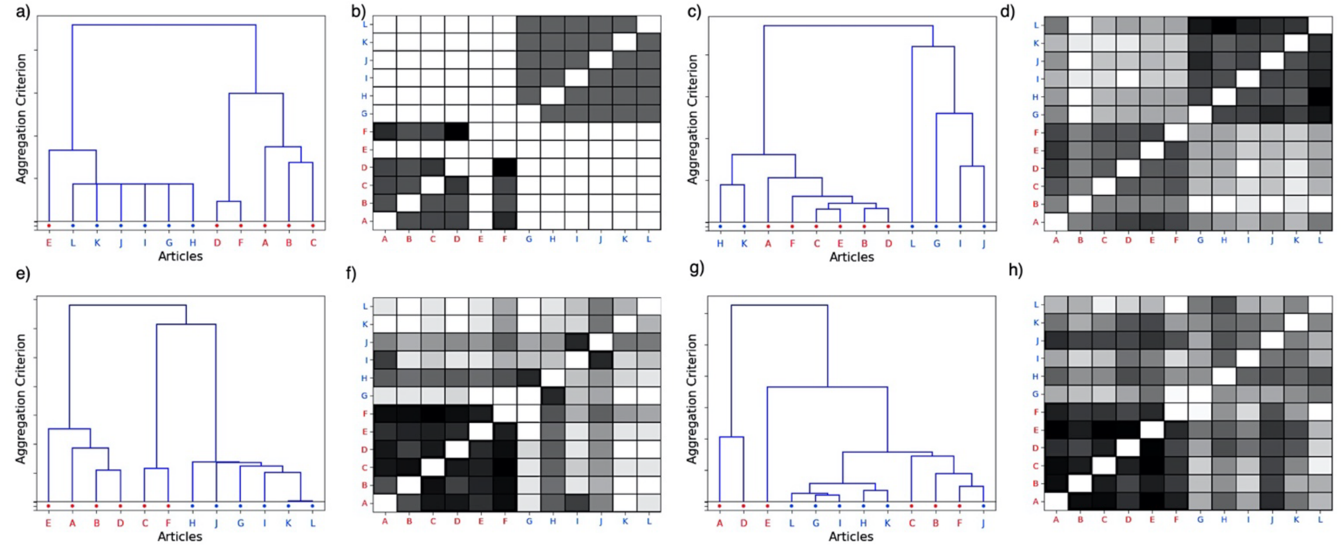

FIGURE 5. Similarity shown on a logarithmic scale in dendrograms, produced using hierarchical agglomerative clustering, and heatmaps. a-d) The two subgroups (red and blue) were generated by running recommend_papers on two different articles. e-h) The two subgroups were generated by searching pairs of distinct but related topics on Google Scholar. Each dendrogram corresponds to the heatmap to its right. Darker color in the heatmaps indicates higher similarity.

Overall, our results suggest that choosing a small number of highly relevant papers and using a small cap can allow researchers to discover a significant fraction of the relevant literature in just a few (<5) iterations of ExCiteSearch. Further, an expert-guided process is likely to be significantly more effective than the semi-random and automated process used in Figure 4.

We also used our similarity metric to perform unsupervised clustering of various sets of papers, using hierarchical agglomerative clustering to create dendrograms. As a simple test, we first generated small datasets by running recommend_papers twice, with a different paper as input each time. The papers were on different topics, and hence this process should create two groups of papers with higher similarity within each group but little similarity between the groups. We then clustered the results using the similarities calculated by a 50-50 mixture of citation connectedness and text similarity (Figure 5a-d). As a more realistic test, we also performed keyword searches on Google Scholar to find papers on two distinct but related topics and extracted the first 6 journal articles on each. Specifically, we searched for two solar cell types (“perovskite solar cell” and “silicon solar cell”, Figure 5e-f) and two catalytic reactions (“ethylene epoxidation” and “propylene epoxidation”, Figure 5g-h). We then performed hierarchical agglomerative clustering on each set of 12 papers (Figure 5a,c,e,g).

As expected, the clustering tends to place more similar papers near each other. While the papers do not perfectly cluster into the two original subgroups, papers from within each subgroup tend to be near each other, and papers that are very closely related show bifurcations low on the dendrogram. The very close relationships in some cases can dominate the clustering, in addition to seeing the rough division into the two input groups.

Overall, hierarchical agglomerative clustering on sets of journal articles allows us to quickly identify groups of related papers, which can give insight into journal scopes, trends in the literature over time, etc. More generally, having a notion of similarity between journal articles can be useful in machine-learning methods that use similarity, such as kernel methods.

IV. CONCLUSION

Research-paper recommender systems are crucial for effectively managing the growing body of scientific literature. Current approaches rely heavily on keyword searching or on a researcher’s entire, curated library. ExCiteSearch’s approach overcomes these limitations by allowing search based on a small set of papers that are known to be relevant and finding related papers using both the citation network and text similarity. ExCiteSearch also allows users to customize the search process and provides multiple modalities for exploring the scientific literature. ExCiteSearch can be freely downloaded at bitbucket.org/mmmontemore/excitesearch.

The effectiveness of ExCiteSearch’s search methods was shown by its ability to reproduce much of the reference list of a review paper and to separate papers on two related topics. With the unique search methods used as well as its efficacy in identifying related papers, the results suggest that ExCiteSearch is quite effective at assembling a comprehensive set of literature on a focused topic.

ExCiteSearch is generally unique among paper recommendation systems due to its use of previously identified papers as inputs and its ability to tailor the search process. While a few tools have recently appeared in this area, ExCiteSearch is distinct from these existing tools in that it is open-source and allows the user to consider both citation connections and textual similarity. ExCiteSearch’s structure avoids many of the pitfalls of keyword searching and recommendations based on an entire library and allows users more controlled and comprehensive exploration of the literature on a focused topic. Having a useful metric of similarity between papers is also useful for informatics and bibliometrics applications. The type of searching we use could be useful in many types of search spaces.

ExCiteSearch may aid researchers in easily finding additional relevant literature to cite, in writing reviews, or in learning about unfamiliar subfields, and its structure is particularly useful for identifying and filling gaps in a tentative reference list. Further improvements in underlying algorithms on text comparison and citation quantification, as well as the speed, robustness, and generality of citation and text retrieval, may further improve the effectiveness and user-friendliness of our framework.

REFERENCES

[1] Z. Jensen, E. Kim, S. Kwon, T. Z. H. Gani, Y. Román-Leshkov, M. Moliner, A. Corma, and E. Olivetti, "A machine learning approach to zeolite synthesis enabled by automatic literature data extraction," *ACS Central Sci.*, vol. 5, no. 5, pp. 892–899, May 2019.

[2] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas, "Music recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 453–492.

[3] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, "Unsupervised word embeddings capture latent knowledge from materials science literature," *Nature*, vol. 571, no. 7763, pp. 95–98, Jul. 2019.

[4] M. C. Swain and J. M. Cole, "ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature," *J. Chem. Inf. Model.*, vol. 56, no. 10, pp. 1894–1904, Oct. 2016.

[5] C.-T. Tsai, G. Kundu, and D. Roth, "Concept-based analysis of scientific literature," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 1733–1738.

[6] L. Bolelli, S. Ertekin, and C. L. Giles, "Clustering scientific literature using sparse citation graph analysis," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, 2006, pp. 30–41.

[7] C. Chen, "Science mapping: A systematic review of the literature," *J. Data Inf. Sci.*, vol. 2, no. 2, pp. 1–40, Mar. 2017.

[8] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific paper recommendation: A survey," *IEEE Access*, vol. 7, pp. 9324–9339, 2019.

[9] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems?: A literature survey," *Int. J. Digit. Libr.*, vol. 17, no. 4, pp. 305–338, 2016.

[10] M. J. Pazzani, "Framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5, pp. 393–408, 1999.

[11] R. Dong, L. Tokarchuk, and A. Ma, "Digging friendship: Paper recommendation in social network," in *Proc. Netw. Electron. Commerce Res. Conf. (NAEC)*, 2002, pp. 1–7.

[12] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10990–11000, Sep. 2012.

[13] L. M. D. Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks," *Int. J. Approx. Reasoning*, vol. 51, no. 7, pp. 785–799, Sep. 2010.

[14] D. D. Nart and C. Tasso, "A personalized concept-driven recommender system for scientific libraries," *Proc. Comput. Sci.*, vol. 38, pp. 84–91, Jan. 2014.

[15] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[16] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Mach. Learn.*, vol. 81, no. 1, pp. 53–67, Oct. 2010.

[17] S. Kataria, P. Mitra, C. Caragea, and C. L. Giles, "Context sensitive topic models for author influence in document networks," in *Proc. Int. Jt. Conf. Artif. Intell. (IJCAI)*, 2011, pp. 2274–2280.

[18] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles, "Context-aware citation recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 421–430.

[19] V. Bush, "As we may think," *Atlantic Monthly*, vol. 176, no. 1, pp. 101–108, Jul. 1945.

[20] J. P. Meneses, "About Pandora and other streaming music services: The new active consumer on radio," *Observatorio*, vol. 6, no. 1, pp. 235–257, 2012.

[21] I. Marshakova-Shaikevich, "System of document connections based on references," *Sci. Tech. Inf. Ser. VINITI*, vol. 6, no. 2, pp. 3–8, Jul. 1973.

[22] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *J. Amer. Soc. Inf. Sci.*, vol. 24, no. 4, pp. 265–269, 1973.

[23] B. Gipp, N. Meuschke, and M. Lipinski, "CITREC: An evaluation framework for citation-based similarity measures based on TREC genomics and PubMed central," in *Proc. iConf.*, 2015, pp. 1–17.

[24] M. Kessler, "Bibliographic coupling between scientific papers," *Amer. Document.*, vol. 14, no. 1, pp. 10–25, 1963.

[25] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf. (SciPy)*, 2008, pp. 11–15.

[26] V. Thada V. Jaglan, "Comparison of Jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm," *Int. J. Innov. Eng. Technol.*, vol. 2, no. 4, pp. 202–205, 2013.

[27] S. Peroni and D. Shotton, "OpenCitations, an infrastructure organization for open scholarship," *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 428–444, Feb. 2020.

[28] R. Lammey, "CrossRef text and data mining services," *Insights UKSG J.*, vol. 28, no. 2, pp. 62–68, Jul. 2015.

[29] Y. Fang, J. A. Powell, E. Li, Q. Wang, Z. Perry, A. Kirchon, X. Yang, Z. Xiao, C. Zhu, L. Zhang, F. Huang, and H.-C. Zhou, "Catalytic reactions within the cavity of coordination cages," *Chem. Soc. Rev.*, vol. 48, no. 17, pp. 4707–4730, 2019.

[30] B. Sun, K. Tao, Y. Jia, X. Yan, Q. Zou, E. Gazit, and J. Li, "Photoactive properties of supramolecular assembled short peptides," *Chem. Soc. Rev.*, vol. 48, no. 16, pp. 4387–4400, 2019.

[31] J. Olesiak-Banska, M. Waszkielewicz, P. Obstarczyk, and M. Samoc, "Two-photon absorption and photoluminescence of colloidal gold nanoparticles and nanoclusters," *Chem. Soc. Rev.*, vol. 48, no. 15, pp. 4087–4117, 2019.

[32] M. Vybornyi, Y. Vyborna, and R. Häner, "DNA-inspired oligomers: From oligophosphates to functional materials," *Chem. Soc. Rev.*, vol. 48, no. 16, pp. 4347–4360, 2019.

[33] S. van der Vorm, T. Hansen, J. M. A. van Hengst, H. S. Overkleeft, G. A. van der Marel, and J. D. C. Codée, "Acceptor reactivity in glycosylation reactions," *Chem. Soc. Rev.*, vol. 48, no. 17, pp. 4688–4706, 2019.

[34] M. Galib, M. D. Baer, L. B. Skinner, C. J. Mundy, T. Huthwelker, G. K. Schenter, C. J. Benmore, N. Govind, and J. L. Fulton, "Revisiting the hydration structure of aqueous Na^+," *J. Chem. Phys.*, vol. 146, no. 8, pp. 1–13, 2017.

[35] K. W. Boyack, C. Smith, and R. Klavans, "A detailed open access model of the PubMed literature," *Sci. Data*, vol. 7, no. 1, pp. 1–16, Dec. 2020.

[36] K. W. Boyack and R. Klavans, "A comparison of large-scale science models based on textual, direct citation and hybrid relatedness," *Quant. Sci. Stud.*, vol. 1, no. 4, pp. 1570–1585, Dec. 2020.

[37] M. Roman, A. Shahid, S. Khan, A. Koubaa, and L. Yu, "Citation intent classification using word embedding," *IEEE Access*, vol. 9, pp. 9982–9995, 2021.

[38] A. Shahid, M. T. Afzal, M. Q. Saleem, M. S. E. Idrees, and M. K. Omer, "Extension of direct citation model using in-text citations," *Comput., Mater. Continua*, vol. 66, no. 3, pp. 3121–3138, 2021.

[39] A. Ruditskiy, H.-C. Peng, and Y. Xia, "Shape-controlled metal nanocrystals for heterogeneous catalysis," *Annu. Rev. Chem. Biomol. Eng.*, vol. 7, no. 1, pp. 327–348, Jun. 2016.

[40] M. S. de Almeida, E. Susnik, B. Drasler, P. Taladriz-Blanco, A. Petri-Fink, and B. Rothen-Rutishauser, "Understanding nanoparticle endocytosis to improve targeting strategies in nanomedicine," *Chem. Soc. Rev.*, vol. 50, no. 9, pp. 5397–5434, 2021.

[41] J. Lv and Y. Cheng, "Fluoropolymers in biomedical applications: State-of-the-art and future perspectives," *Chem. Soc. Rev.*, vol. 50, no. 9, pp. 5435–5467, 2021.

[42] H. Wang, M. Wang, X. Liang, J. Yuan, H. Yang, S. Wang, Y. Ren, H. Wu, F. Pan, and Z. Jiang, ''Organic molecular sieve membranes for chemical separations,'' *Chem. Soc. Rev.*, vol. 50, no. 9, pp. 5468–5516, 2021.

[43] D. Aynetdinova, M. C. Callens, H. B. Hicks, C. Y. X. Poh, B. D. A. Shennan, A. M. Boyd, Z. H. Lim, J. A. Leitch, and D. J. Dixon, ''Installing the 'magic methyl'—C–H methylation in synthesis,'' *Chem. Soc. Rev.*, vol. 50, no. 9, pp. 5517–5563, 2021.

[44] R. Gao, M. S. Kodaimati, and D. Yan, ''Recent advances in persistent luminescence based on molecular hybrid materials,'' *Chem. Soc. Rev.*, vol. 50, no. 9, pp. 5564–5589, 2021.

[45] J. Huo and B. H. Shanks, ''Bioprivileged molecules: Integrating biological and chemical catalysis for biomass conversion,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 11, no. 1, pp. 63–85, Jun. 2020.

[46] P. Marchetti, L. Peeva, and A. Livingston, ''The selectivity challenge in organic solvent nanofiltration: Membrane and process solutions,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 8, no. 1, pp. 473–497, Jun. 2017.

[47] V. P. Chauhan, T. Stylianopoulos, Y. Boucher, and R. K. Jain, ''Delivery of molecular and nanoscale medicine to tumors: Transport barriers and strategies,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 2, no. 1, pp. 281–298, Jul. 2011.

[48] D. L. Keairns, R. C. Darton, and A. Irabien, ''The energy-water-food Nexus,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 7, no. 1, pp. 239–262, 2016.

[49] R. C. Flagan, ''Continuous-flow differential mobility analysis of nanoparticles and biomolecules,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 5, no. 1, pp. 255–279, Jun. 2014.

[50] A. Klamt, F. Eckert, and W. Arlt, ''COSMO-RS: An alternative to simulation for calculating thermodynamic properties of liquid mixtures,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 1, no. 1, pp. 101–122, Jun. 2010.

[51] N. Fackler, B. D. Heijstra, B. J. Rasor, H. Brown, J. Martin, Z. Ni, K. M. Shebek, R. R. Rosin, S. D. Simpson, K. E. Tyo, R. J. Giannone, R. L. Hettich, T. J. Tschaplinski, C. Leang, S. D. Brown, M. C. Jewett, and M. Köpke, ''Stepping on the gas to a circular economy: Accelerating development of carbon-negative chemical production from gas fermentation,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 12, no. 1, pp. 439–470, Jun. 2021.

[52] E. Bukusoglu, M. Bedolla Pantoja, P. C. Mushenheim, X. Wang, and N. L. Abbott, ''Design of responsive and active (soft) materials using liquid crystals,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 7, no. 1, pp. 163–196, Jun. 2016.

[53] B. R. Bakshi, ''Toward sustainable chemical engineering: The role of process systems engineering,'' *Annu. Rev. Chem. Biomol. Eng.*, vol. 10, no. 1, pp. 265–288, Jun. 2019.

JULIA A. STERLING is currently pursuing the B.S. degree in mathematics and linguistics with Tulane University. She has been working with research-paper recommender systems, since 2019. Her research interests include recommender systems, databases, and number theory.

MATTHEW M. MONTEMORE received the B.A. degree in physics from the Grinnell College, the M.S. and Ph.D. degrees in mechanical engineering from the University of Colorado Boulder, and performed postdoctoral research in chemistry at Harvard University. He is currently a Robert and Gayle Longmire Early Career Professor with the Department of Chemical and Biomolecular Engineering, Tulane University. His research interests include applying data science to materials and chemical applications, extracting information from scientific literature, and computational studies of catalysts and other materials for energy applications.