

Entrup, Elias et al.

Article — Published Version

Comparing different search methods for the open access journal recommendation tool B!SON

International Journal on Digital Libraries

Provided in Cooperation with:

Springer Nature

Suggested Citation: Entrup, Elias et al. (2023) : Comparing different search methods for the open access journal recommendation tool B!SON, International Journal on Digital Libraries, ISSN 1432-1300, Springer, Berlin, Heidelberg, Vol. 25, Iss. 3, pp. 505-516, <https://doi.org/10.1007/s00799-023-00372-3>

This Version is available at:

<https://hdl.handle.net/10419/309010>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



Comparing different search methods for the open access journal recommendation tool B!SON

Elias Entrup¹ · Anita Eppelin¹ · Ralph Ewerth^{1,2} · Josephine Hartwig³ · Marco Tullney¹ · Michael Wohlgemuth³ · Anett Hoppe^{1,2}

Received: 11 November 2022 / Revised: 25 May 2023 / Accepted: 25 May 2023 / Published online: 20 July 2023
© The Author(s) 2023

Abstract

Finding a suitable open access journal to publish academic work is a complex task: Researchers have to navigate a constantly growing number of journals, institutional agreements with publishers, funders' conditions and the risk of predatory publishers. To help with these challenges, we introduce a web-based journal recommendation system called B!SON. A systematic requirements analysis was conducted in the form of a survey. The developed tool suggests open access journals based on title, abstract and references provided by the user. The recommendations are built on open data, publisher-independent and work across domains and languages. Transparency is provided by its open source nature, an open application programming interface (API) and by specifying which matches the shown recommendations are based on. The recommendation quality has been evaluated using two different evaluation techniques, including several new recommendation methods. We were able to improve the results from our previous paper with a pre-trained transformer model. The beta version of the tool received positive feedback from the community and in several test sessions. We developed a recommendation system for open access journals to help researchers find a suitable journal. The open tool has been extensively tested, and we found possible improvements for our current recommendation technique. Development by two German academic libraries ensures the longevity and sustainability of the system.

Keywords Recommendation system · Open access · Paper submission · Academic publishing · Scholarly journals

1 Introduction

The open access landscape keeps getting more diverse and complex. An increasing number of open access journals make it harder to choose a suitable journal for a particular research output: The Directory of Open Access Journals (DOAJ), a curated online directory of peer-reviewed open access journals, added over 5,000 journals in the last three years. All of these journals offer a variety of conditions, including different publication costs and waivers, peer-review models, and copyright and rights retention clauses. At the same time,

a growing number of funding agencies require researchers to publish open access, in compliance with specific rules. Academic libraries increasingly offer financial support, also introducing new criteria. All of these developments add to the overall workload of researchers who now have to choose proper publication venues and to assess them in terms of compliance, publication costs, quality, and reputation.

In this paper, we present a detailed look into B!SON, a web-based recommendation system that aims to alleviate these issues. The system uses basic information from the manuscript to be published to recommend suitable open access journals, based on content similarity (using title and abstract) and the cited references. The initial design of the B!SON service is based on the findings of a survey [1] conducted at the start of the project. Therein, we systematically collected user requirements which were directly integrated into the system specification.

This article is an extended version of a short paper [2] presented at the TPD 2022. Apart from more details on the user survey (Sect. 3.1), information on additional B!SON services

✉ Elias Entrup
bison@tib.eu

¹ TIB-Leibniz Information Centre for Science and Technology, Welfengarten 1B, 30167 Hannover, Germany

² Leibniz University Hannover, L3S Research Center, Appelstr. 9a, 30167 Hannover, Germany

³ Saxon State and University Library, Zellescher Weg 18, 01069 Dresden, Germany

(Sect. 3.5) and a discussion (Sect. 5), it includes new results from the ongoing development process: Sect. 3.3.2 describes embedding-based approaches we explored to improve the semantic component of the current recommendation system. They have been implemented and subjected to a comparative evaluation of their recommendation performance (Sect. 4). An already implemented and integrated extension to the graphical user interface, filter suggestions, is explained in Sect. 3.4.2.

The paper is structured as follows: We first provide a review of existing work on scientific recommendation (Sect. 2). Subsequently, we present the B!SON prototype and its development: Sect. 3.1 discusses the initial assessment of user requirements; Sect. 3.2 presents the integrated data sources; Sect. 3.3 explains the current recommendation algorithm as well as currently tested, more advanced versions. In Sects. 3.4 and Sect. 3.5, we provide details on the system's user interface and the planned TYPO3 extension for local instances of the service, respectively. The results of the experimental evaluation of different recommendation methods are presented in Sect. 4. A discussion (Sect. 5) and conclusion (Sect. 6) follow.

2 Related work

Scientific recommendation tasks span the search for potential collaborators [3] and reviewers [4], of papers to read [5] and to cite [6]. With more and more ways of publishing academic articles, the recommendation of academic publication outlets (journals/conferences) is a task which is on the rise (e.g., [7, 8])

Prototypical approaches explore diverse data sources to provide recommendations. A major source of information is the article to be published: The manuscript's title, abstract or keywords are used to compare against papers that previously appeared in an outlet [8, 9]. Other systems exploit the literature cited by the article, and try to determine the best publication venue using bibliometric methods [10, 11]. An alternative stream of research focuses more on the article authors, exploring their publication history [12] and co-publication networks [13, 14].

Regarding journal recommendation based on semantic similarity, TF-IDF is a popular building block [15–17], especially in combination with chi-square statistics to determine the dependence of terms compared to journals [18–20]. Other systems use a word embedding like word2vec or fasttext in combination with a convolutional neural network [21–23]. Algorithms of popular search engines like Okapi BM25 or MoreLikeThis are used in [24, 25]. Others use document-level embeddings [26, 27], approaches based on n-grams [28] or manually defined ratios [29]. Systems using algorithms that do not directly return journal recommendations,

but a set of similar articles, rely on aggregation methods, e.g., using the k-Nearest-Neighbors algorithm in combination with summation or averaging to calculate a journal score [24, 25].

Semantic recommendation is usually based on title, abstract and keywords [15, 16, 20, 22, 23]. The Aims & Scope section of the journal can be considered as well [17, 30]. The references section might be used [26, 29] as well as the full text including images [31].

While there is a number of active journal recommender sites, they all come with limitations. Several publishers offer services limited to their own journals like Elsevier's Journalfinder¹ or Springer's Journal suggester.² Others, like Journal Guide,³ are closed-source and do not provide transparent information on their recommendation approach. Several services collect user and usage data, e.g., Web of Science's manuscript matcher.⁴ These proprietary services work with semantic methods, recommending journals based on title, abstract and keywords. Notably, some open recommenders exist, e.g., Open Journal Matcher,⁵ Pubmender,⁶ Jane⁷ and Jot.⁸ Since the publication of the first version of this article, the author of the Open Journal Matcher announced that the service will be discontinued [32] and the Pubmender back end does not seem to work anymore. The recommendations of Jane and Jot are limited to medical journals. These open services integrate little information about the recommended journals and do not offer advanced filter options.

3 B!SON—the open-access journal recommender

B!SON is the abbreviation for *Bibliometric and Semantic Open Access Recommender Network*. It combines several available data sources to provide authors and publication support services at libraries with recommendations of suitable open access journals, based on the title, abstract and reference list of the paper to be published. The system will be maintained for at least five years after which its usage will be evaluated. Further extensions of the core functionality are planned.

¹ <https://journalfinder.elsevier.com/>.

² <https://journalsuggester.springer.com/>.

³ <https://www.journalguide.com/>.

⁴ <https://mjl.clarivate.com/manuscript-matcher>.

⁵ <https://ojm.ocert.at/>.

⁶ <https://www.keaml.cn/abc/>.

⁷ <https://jane.biosemantics.org/>.

⁸ <https://jot.publichealth.yale.edu/>.

3.1 Survey

To assess the needs of the future B!SON users, we conducted an online survey as a requirements analysis [33]. The question items were based upon features of existing journal recommender tools; the survey was aimed at scientists from all research disciplines.

After discarding entries with less than 90% questions answered, a total set of 884 questionnaires remained for analysis. The participating researchers had an experience of having published a median of seven papers.

The survey targeted two main categories of information: (a) finding out which filter functionalities are most important for efficiently selecting a journal; (b) assessing the key characteristics of journals to display in a journal profile. The results differ only slightly depending on the research discipline. Overall, the most important filter criteria were citation metrics, publication costs, language of the publication, appearance in scholarly databases and whether the author retains the copyright. For the journal characteristics to show, the most important information is whether the article receives a DOI, whether the journal is listed in common journal lists to protect against predatory publishers, whether the publication costs are covered, the journal's scope and general publication costs.

We considered these results in the design of B!SON where possible while also keeping the focus on few and trusted data sources. Citation metrics were deliberately not included in the design due to their controversial influence [34].

3.2 Data sources & integration

The B!SON service is built on top of several open data sources with strong reputation in the open access community:

DOAJ: The Directory of Open Access Journals (DOAJ)⁹ indexes information on open access journals which fulfil a set of quality criteria (full text available, dedicated article URLs, at least one ISSN etc.) as well as publishing ethics guidelines. The dataset includes basic information on the journal itself, but also metadata of the published articles (title, abstract, year, DOI, ISSN of journal, etc.). The DOAJ currently contains 18,461 journals and 8,154,699 articles. The data are available for download in JSON format under CC0 for articles and CC BY-SA for journal data [35].

OpenCitations: The OpenCitations initiative¹⁰ collects (amongst others) the CC0-licensed COCI dataset for citation data. It is based on Crossref data and contains 76,072,926 publications, and 1,392,036,835 citations [36]. The information is available in the form of DOI-to-DOI relations,

covering 44% of citations in Scopus and 51% of the citations in Dimensions [37]. COCI lacks citations in comparison with commercial products but can be used to check which articles published in DOAJ journals cite the references given by the user (details in Sect. 3.3.1). The coverage of open access publications, especially DOAJ journals, in COCI is better compared to closed access publications, so we can assume that it is sufficient for our needs [38].

Journal Checker Tool: The cOAlition S initiative (a group of funding agencies that agreed on a set of principles called Plan S for the transition to open access) provides the Journal Checker Tool.¹¹ A user can enter journal ISSN, funder and institution to check whether (a) a journal is fully open access according to Plan S requirements, (b) the journal is a transformative journal, (c) it has a transformative agreement with the user's institution, or (d) the journal offers a self-archiving option [39]. An API allows fetching this information automatically. Since B!SON does not retrieve data on the funder or institution, and the DOAJ dataset only contains open access journals (and no transformative journals), we use the funder information of the European Commission as a placeholder to check if a journal is Plan-S compliant.

Additional data: There are other data sources which might be used in future B!SON versions to extend the current setup. Crossref metadata would allow us to extend the article data of the DOAJ which are occasionally incomplete. OpenAlex¹² could add, e.g., author information.

Data integration: Data from DOAJ and OpenCitations' COCI index are bulk downloaded and inserted into two databases: PostgreSQL and Elasticsearch. The information on Plan-S compliance stems from the Journal Checker Tool and is fetched from the API using the "European Commission Horizon Europe Framework Programme" as placeholder for the funder. The DOAJ articles are matched to their journal via ISSN, and matching to the citations happens via DOI. Data on Plan S compliance are connected via ISSN as well.

All software is published and developed as open source under the AGPL licence on GitLab.¹³ The used data sources are automatically updated in regular intervals to keep the service up-to-date without human intervention. For transparency, the time of the last update is shown on B!SON's "About" page.

3.3 Recommendation system

B!SON consists of a Django back end¹⁴ and a Vue.js¹⁵ front end. The original publication presented an implementation

⁹ <https://doaj.org>.

¹⁰ <https://opencitations.net>.

¹¹ <https://journalcheckertool.org/>.

¹² <https://openalex.org/>.

¹³ <https://gitlab.com/TIBHannover/bison/>.

¹⁴ <https://www.django-project.com/>.

¹⁵ <https://vuejs.org/>.

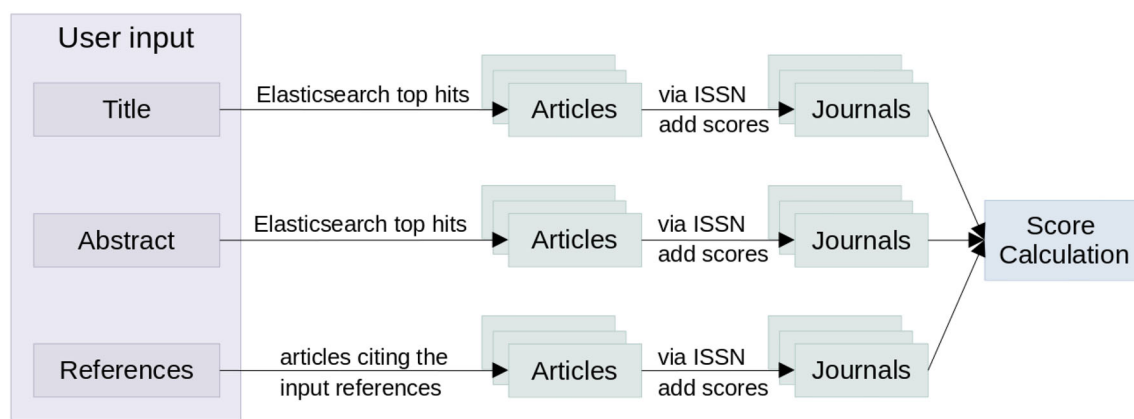


Fig. 1 Flow diagram showing how articles similar to the user input are found and then matched to their journal. The score for each journal is calculated as a final step

using a basic recommendation algorithm (here described in Sect. 3.3.1). Since then, we experimented with a number of possible enhancements using embedding-based approaches which are presented in Sect. 3.3.2. The comparative evaluation follows in Sect. 4.

3.3.1 Baseline system

The current recommendation system is based on combined similarity measures with regard to the entered text data (title and abstract) and reference list. Figure 1 shows an overview of the recommendation process, the individual steps are described in the following passages.

Text similarity: Elasticsearch has a built-in functionality for text similarity search based on the Okapi BM 25 algorithm [40]. This functionality is used to determine those articles already indexed in the DOAJ which are similar to the entered information. Stop word removal is performed as a pre-processing step. The DOAJ contains articles in several languages, so we combine the available Apache Lucene stop word lists¹⁶ for this purpose. The similarity search happens separately for title and abstract and only the top 100 hits are considered.

Bibliographic coupling: According to Kessler [41], two articles are *bibliographically coupled* if at least one reference is cited in both articles. This linking citation can be interpreted as an overlap in content or method. The more references are cited together, the higher the closeness of the two articles [42].

A *co-citation* exists if two articles are cited together in a third article. Co-citation, too, can be assumed to indicate article similarity. Figure 2 shows the temporal dependence of both methods. While co-citation calculates the similarity of

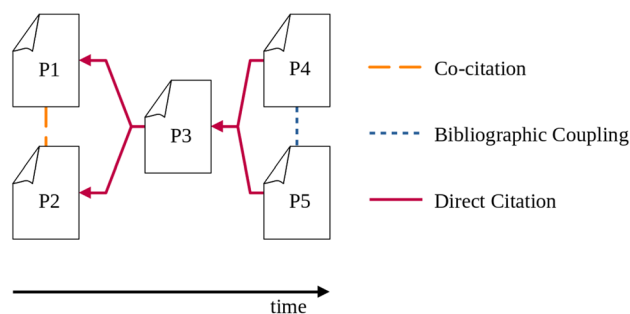


Fig. 2 Visualization of the relation between bibliographic coupling, co-citation, and direct citations (adapted from [43])

already cited and, thus, older papers, bibliographic coupling can be used to calculate the proximity of recent articles.

This approach of a similarity calculation of recent articles is inspired by the process of the publication support services of one of the participating libraries. It is currently used in B!SON in the following way:

The user enters an unstructured list of references which are cited in the article to be matched to a journal. From this list, B!SON extracts the DOIs using regular expressions. Then, it relies on OpenCitations' COCI index to find existing articles citing the same sources. The current normalization of the degree of bibliographic coupling is in a prototypical state: The number of matching citations is divided by the highest number of matching citations of the compared articles. If this normalized value is higher than a threshold (which is currently manually defined), the article is considered similar. The system then determines the journals in which the similar articles have been published, taking only those into account which are indexed in the DOAJ. The more articles in a journal were considered relevant based on bibliographic coupling, the higher the respective journal will rank in the generated result list.

¹⁶ <https://github.com/apache/lucene/tree/main/lucene/analysis/common/src/resources/org/apache/lucene/analysis>.

Combination of text-based and bibliographic similarity: Similar articles are matched with their journal. The maximum of three different scores per journal are combined using a neural network which was trained to classify a journal as correct or incorrect based on these scores, thereby weighing them in their meaningfulness. The resulting probability is the output score for each journal which will then be displayed as part of the result page. To increase transparency of the scoring, we are investigating alternatives to this combination process.

3.3.2 Embedding-based approaches

To further improve the recommendations, we tested several other approaches. Existing recommendation systems often use language models to build text representations [21]. The journals referenced in the DOAJ include articles in multiple languages, raising the requirement for a multilingual language model. This eliminates options such as commonly used transformer-based models for scientific language (e.g., SciBERT [44]). We tested three different multilingual language models (see Sect. 4.2.2).

One approach is to find similar articles with vector embeddings instead of word comparisons (like Elasticsearch). Each article's title and abstract are combined and fed into the pre-trained, transformer-based language model. The resulting embeddings can then be compared against the embedding of the user input. Previous work also suggested a more fine-grained approach, weighing journal articles lower for the journal embedding if they appeared earlier in the past [45]. The distribution of our dataset features many journals with few articles (see Sect. 5), and does not allow for this kind of granularity.

The embeddings can be compared on the article level and on the journal level. We experiment with the following configurations.

Journal embeddings: For the journal level, the article embeddings of each journal are combined by calculating the average. The embeddings that are the closest to the embedding of the input text are the best matching journals. The Open Journal Matcher uses a similar approach (using spacy¹⁷ as a word-level embedding model instead of a document embedding).

Article embeddings (individual): For the article level, the embedding of the user input is directly compared to all article embeddings. The rank of a journal is derived from its article with the closest embedding.

Article embeddings (combined): Similarly, the embedding of the user input is directly compared to all article embeddings. Only the articles within the top-n hits are used

for the computation of the combined journal score, which is calculated by summing up the scores.

Article embedding & classifier: Building upon existing pre-trained, transformer-based language models, a classifier can be trained to predict a journal. The weights of the pre-trained model are frozen and a dense layer is added to predict the one-hot encoded journals based on the classification token of the pre-trained model.

Article embedding, BiLSTM & classifier: The previous approach can be further refined by adding a BiLSTM layer in-between that receives the token embeddings of the pre-trained language model as an input.

We created the embeddings using the Sentence-Transformers library¹⁸ which offers the functionality to include and test different language models. The gensim library¹⁹ provides the functionality to determine the closest neighbors using the dot product on normalized vectors. We used the Huggingface library²⁰ in combination with PyTorch²¹ to try different pre-trained, transformer-based and multilingual models. The weights of the language model were not further finetuned. The Huggingface tokenizer of the corresponding model was used with a token length of 256. For the BiLSTM layer, 256 hidden features were used. The Universal-Sentence-Encoder model was obtained from TensorFlow Hub²² and the classification layer was trained with TensorFlow. The results are shown in Table 3.

3.4 Interfaces & functionality

The current state of the B!SON system is available online.²³ The main entry point for an end user is the graphical user interface provided on our website; it is described in Sect. 3.4.1. It was recently extended by support functionalities for data entry (Sect. 3.4.2). Beyond that, we provide additional access points for programmatic access and integration into third-party services (Sect. 3.5).

3.4.1 Graphical user interface

The user interface has been designed deliberately simple; screenshots are shown in Figs. 3 and 4.

Data entry: The start page allows the user either to enter title, abstract and references directly or to let them be filled out automatically by fetching the information from Crossref, DataCite or arXiv with a DOI or arXiv ID. This allows open

¹⁷ <https://spacy.io/>.

¹⁸ <https://www.sbert.net/>.

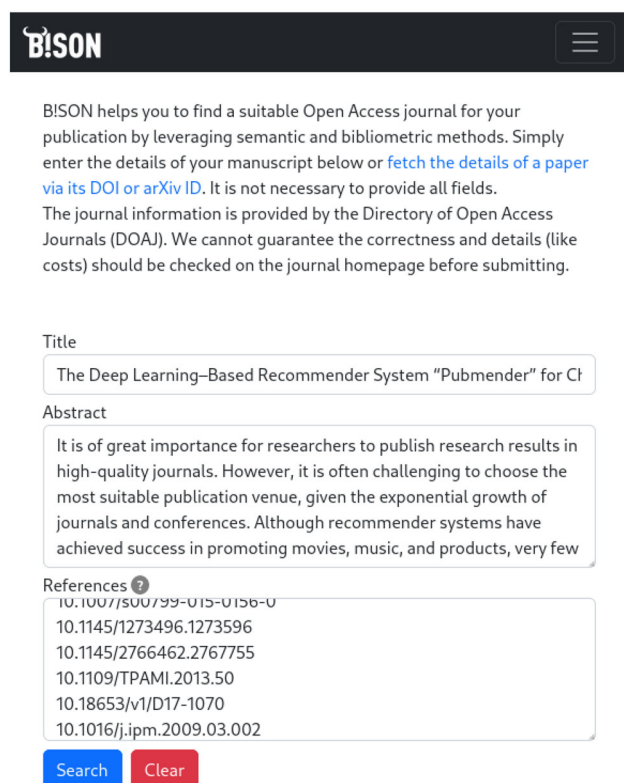
¹⁹ <https://radimrehurek.com/gensim/>.

²⁰ <https://huggingface.co/>.

²¹ <https://pytorch.org/>.

²² <https://tfhub.dev/google/collections/universal-sentence-encoder/1>.

²³ <https://service.tib.eu/bison/>.



B!SON

B!SON helps you to find a suitable Open Access journal for your publication by leveraging semantic and bibliometric methods. Simply enter the details of your manuscript below or [fetch the details of a paper via its DOI or arXiv ID](#). It is not necessary to provide all fields. The journal information is provided by the Directory of Open Access Journals (DOAJ). We cannot guarantee the correctness and details (like costs) should be checked on the journal homepage before submitting.

Title
The Deep Learning-Based Recommender System "Pubmender" for CI

Abstract
It is of great importance for researchers to publish research results in high-quality journals. However, it is often challenging to choose the most suitable publication venue, given the exponential growth of journals and conferences. Although recommender systems have achieved success in promoting movies, music, and products, very few

References ?
10.1007/978-3-319-0156-0
10.1145/1273496.1273596
10.1145/2766462.2767755
10.1109/TPAMI.2013.50
10.18653/v1/D17-1070
10.1016/j.jipm.2009.03.002

Search Clear

Fig. 3 Screenshot of B!SON prototype with an example query

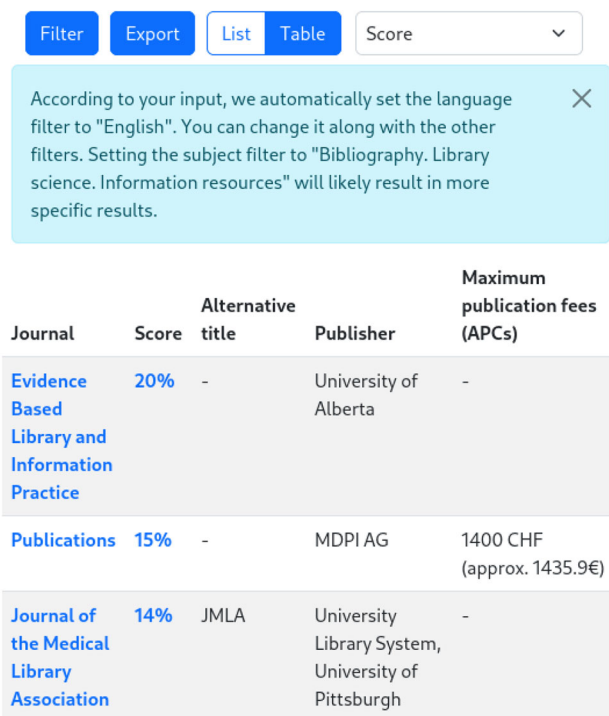
access publication venues to be found based on previously published research.

Results page: To inspect the search results, the user has the choice of representing them either as a simple list or a table which offers a structured account of additional details, enabling easy comparison of the journals. Article processing charges (APCs) are displayed based on the information available in DOAJ, and automatically converted to Euro if necessary.

After clicking on the score field, a pop-over with explanatory information displays a list of articles which previously appeared in said journal which were determined to be similar by the recommendation engine. Clicking on a journal title leads the users to a separate detail page which offers even more information including keywords, APCs, licence, Plan-S compliance, and more.

3.4.2 Data entry support

The B!SON website offers the possibility to refine the results with a number of filter options. While several of them are purely preferences by the user (such as the average publication time), the language and subject can be deduced from the user input and presented as a suggestion.



Filter Export List Table Score

According to your input, we automatically set the language filter to "English". You can change it along with the other filters. Setting the subject filter to "Bibliography. Library science. Information resources" will likely result in more specific results.

Journal	Score	Alternative title	Publisher	Maximum publication fees (APCs)
Evidence Based Library and Information Practice	20%	-	University of Alberta	-
Publications	15%	-	MDPI AG	1400 CHF (approx. 1435.9€)
Journal of the Medical Library Association	14%	JMLA	University Library System, University of Pittsburgh	-

Fig. 4 Screenshot of B!SON prototype showing the table view with results

Language: We are using the library `lingua-py`²⁴ to choose the most likely language for the given input text.

Subject: A neural network is used to identify the subject. We adopted the Library of Congress Classification (LCC) which is used by the DOAJ and constructed a training set of 10,000 DOAJ articles for each top level subject (based on the first letter of the subject code). The pretrained multilingual language model XLM- ROBERTA- LARGE [46] with a BiLSTM and classification layer as a head was trained with an accuracy of 73.56% on a test set of 1000 articles per subject.

The model XLM- ROBERTA [46] is a transformer based multilingual language model trained with the Masked-Language-Modeling objective. The training set consists of a subset of Common Crawl with over 100 languages.

Semantically close categories within the LCC such as "World History and History of Europe, Asia, Africa, Australia, New Zealand, etc." and "Auxiliary Sciences of History" can pose a challenge to our trained model. The subject suggestion is only presented to the user if a probability greater than 50% for the prediction is reached.

²⁴ <https://github.com/pemistahl/lingua-py>.

3.5 Further interfaces

To enhance integration and computational interoperability, other access options apart from the standard user interface are provided.

Data export: Search results can be exported as CSV for further sharing and analysis.

API: A public API is available for programmatic access. As it is also used for the front-end, all shown information on the website can be accessed via the API.

Local instances: We plan to provide the recommendation functionality in a form that can be easily integrated and adapted to third-party websites, e.g., by libraries. We are currently starting with the development of an extension for the TYPO3 Content Management System (CMS)²⁵ which is widely used in the German library landscape. Both the TIB and the SLUB library are using TYPO3 and will act as early adopters of the prototype. The extension will allow libraries to further filter the results or include additional information such as waiver agreements with publishers. The code of the extension will be provided as open source for re-use and support for other content management systems is planned.

4 Experimental evaluation

The recommendation algorithms were tested in two different experimental setups (Sect. 4.1); Sect. 4.2 discusses the achieved results.

4.1 Experimental setups

The algorithms were tested with two different setups. The first one uses a randomly sampled dataset from the DOAJ data and consequently simulates an article distribution corresponding to the article distribution in the full dataset. This entails a higher share of data coming from certain academic domains and/or journals. This data setting thus represents the realistic environment in which our prototype needs to perform.

For the second setup, we sampled a fixed number of articles from all eligible journals in the DOAJ. As it is not skewed to certain domains and journals, it allows for a fairer comparison of the enhanced methods on journal and article level.

4.1.1 Random article sampling

The algorithm is evaluated on a separate test dataset of 10,000 random DOAJ articles. To ensure realistic input data, all articles in the test set have a minimal abstract length of 150 characters and a minimal title length of 30 characters. As the

Table 1 Requirement levels for title length, abstract length and number of references. The last column lists the number of journals which have the required amount of complying articles

Level	Title	Abstract	References	Journals
Low	0	0	0	12,112
Middle	50	650	15	3083
High	100	1300	30	1976

references are not part of DOAJ's article metadata, the COCI index was used to complete the references via the article DOI. Only articles with at least five references were included. We assume that the articles were published in a suitable journal to begin with, counting a positive result if the originating journal appears in the top-n results of the recommendation. While this may not be correct for each individual article, we rely on the assumption that the overall journal scope is defined by the articles published in that journal.

Random sampling introduces a bias. Subjects such as medicine have a higher share than other subjects. This limitation of the dataset is further discussed in Sect. 5.

4.1.2 Equal article distribution

The distribution of articles in the DOAJ is skewed toward certain domains and journals. To provide a fairer test case, we sample a subset of DOAJ articles: For each journal, 20 random articles are included in the training set; 10 articles per journal go to the test set. We then proceed to a detailed analysis concerning the influence of minimal text length of the input data and the number of references found in COCI on the overall recommendation accuracy. We define three levels of requirements as to these parameters, as shown in Table 1. The "high" requirement values correspond to the median in the dataset.

The articles' title and abstract were pre-processed to remove HTML entities, URLs and non UTF-8 characters. As the bibliometric recommendation works with a different approach than the semantic ones, the data model has to be slightly changed to create a suitable comparison: after finding citations via COCI only those publications are matched to their DOAJ journal whose DOI is within the training data. This simulates the same case of 20 articles per journal that the semantic comparison assumes.

4.2 Experimental results

This section discusses the results of the experimental evaluation in the two setting discussed in Sect. 4.1.

²⁵ <https://typo3.org/>.

Table 2 Top@N accuracy for the current recommendation tested on 10,000 random articles

Method	Top@1	Top@5	Top@10	Top@15
Bibliometric approach	0.242	0.510	0.631	0.692
Elasticsearch on title	0.205	0.452	0.563	0.620
Elasticsearch on abstract	0.261	0.533	0.646	0.700
Combined approach	0.269	0.562	0.687	0.749

4.2.1 Random article sampling

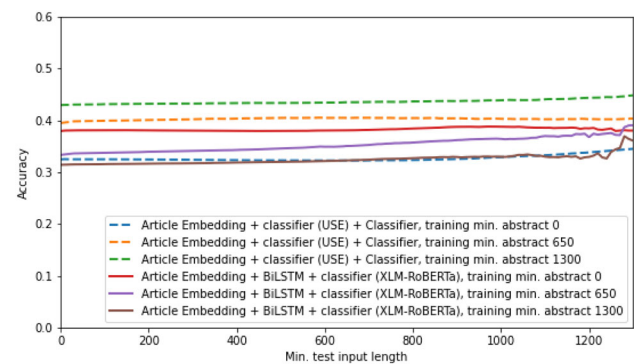
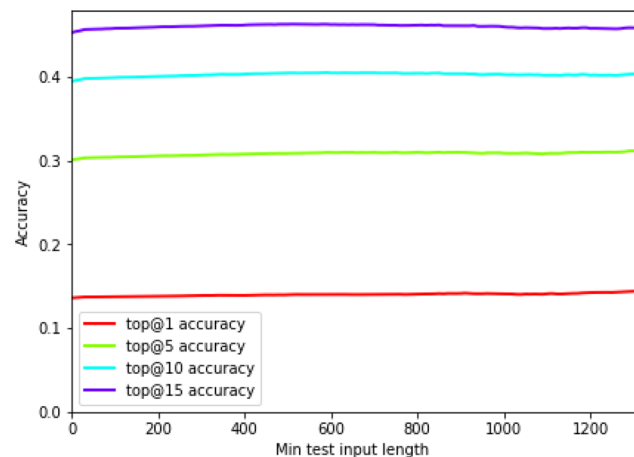
Here, we discuss the results of the evaluation setting described in Sect. 4.1.1. Table 2 shows the results for the top@N accuracy for using (a) the bibliometric approach alone (“Bibliometric approach”); (b) only the Elasticsearch provided similarity score for the title (“Elasticsearch on title”) (c) only the Elasticsearch provided similarity score for the abstract (“Elasticsearch on abstract”) (d) using the recommendation combining all of the above (“combined approach”) which is the solution used in the current version of the B!SON system. Unsurprisingly, the abstract-based recommendation delivers better results than the title-based approach, as the former provides more information. The combination of the recommendation methods shows the expected improvement over the individual methods.

4.2.2 Equal article distribution

Here, we discuss the results related to the experimental setup discussed in Sect. 4.1.2. Again included are the algorithms constituting the current solution, named as in Sect. 4.2.1 and Table 2. Additionally, we report the results for the embedding-based approaches discussed in Sect. 3.3.2.

Impact of training data: The effect of the minimal training requirements is shown in Fig. 5. As expected, the accuracy during testing improves if longer texts are used for training. This was the case for all methods except for XLM-RoBERTa with a classifier or BiLSTM and classifier layer (shown for the latter case in the same plot). The reason is unclear and requires further investigation. Possible reasons include that model learns undesired features such as assigning short texts generally to a subset of journals.

Impact of test data: We evaluated the models’ top@1, top@5, top@10 and top@15 accuracy, varying the length of the input data used for testing. (The model was trained with input data that satisfy the “middle” requirements criterion described above.) the models exhibit similar behavior, we thus show only one example of the resulting graph in Fig. 6, using the USE-based system configuration. In general, there is a bigger leap from the top@1 accuracy to top@5 accuracy, while the gap between top@5 and top@10 as well as top@10 and top@15 is decreasing in size.

**Fig. 5** The top@10 accuracy for the “USE article embedding with classifier” and “XLM-RoBERTa + BiLSTM + classifier” method with respect to different requirements on the training set**Fig. 6** The different accuracies for the “USE article embedding with classifier” method for the “middle” requirements for the training set

The accuracy for the different methods is presented in Fig. 7. To reduce the visual complexity, only the top@10 accuracy is shown. Both the Universal-Sentence-Encoder journal vectors and the trained classifier are performing well. In contrast to XLM-RoBERTa, the USE model only supports 16 languages and does not cover all languages from DOAJ corpus. The accuracy of the Elasticsearch recommendation on the abstract increases almost linearly with longer input length. For some methods (e.g., journal embedding with Universal-Sentence-Encoder) the line levels out at some point. This is caused by the models’ limitations w.r.t. maximum input lengths.

The results are shown in detail in Table 3 for the “middle” requirements level applied to both training and test set. As highlighted in the table, the best performing algorithms for this configuration are the bibliometric search and the Universal-Sentence-Encoder journal embeddings. The version with XLM-RoBERTa in combination with a BiLSTM is on fourth place regarding top@10 and top@15 accuracy. It is the method that we identified as well performing in pre-

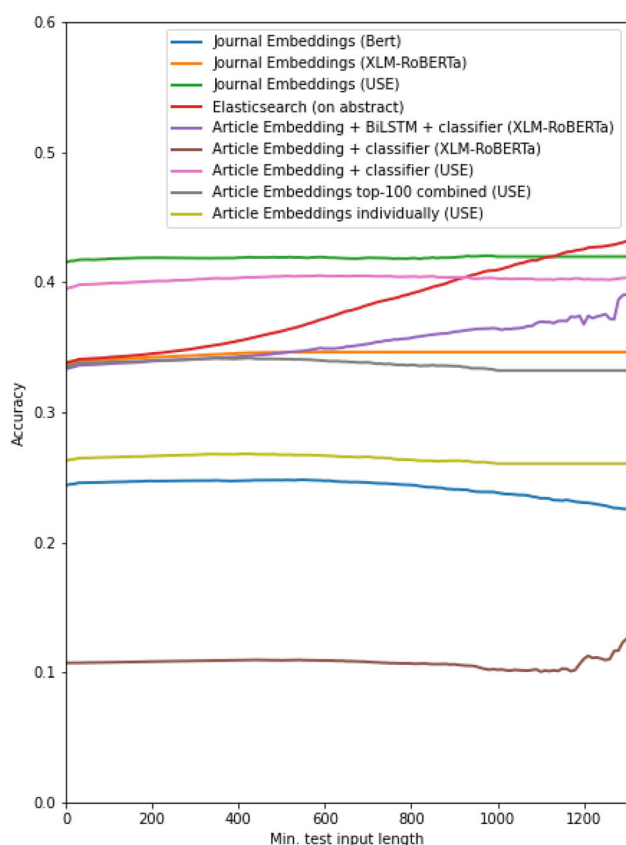


Fig. 7 The top@10 accuracy for the different methods presented with the “middle” requirements for the training set

vious tests and that is used for subject prediction (described in Sect. 3.4.2).

We further report training and testing times in Table 4. For the Elasticsearch approach, the indexing time (for both title and abstract at once) was measured as the training time. The experiments were conducted on a server with an “AMD EPYC 7542” processor, NVMe connected hard drives and “AMD A40” graphic cards. Elasticsearch and the bibliometric search take the longest time as they require many database look-ups.

5 Discussion

While the current B!SON implementation achieves a decent accuracy and beta users reported predominantly positive feedback, the approach comes with a set of limitations. Using the data from the DOAJ has the advantage of reliable data access and basic quality control of the metadata as well as ensuring a minimal standard of journal’s publishing ethics policies. However, information may be incomplete or not up-to-date. For instance, information such as Article Processing Charges (APCs) are provided by publishers once, and might

not be adjusted over time. Furthermore, the calculation of the exact APCs is sometimes rather complex, with charges influenced by the number of pages or figures in a manuscript – a fact which cannot be considered in the B!SON interface. Displayed APC information are thus only indicative.

Moreover, the distribution of articles is highly skewed. There are few journals with an immense number of articles (4 with more than 50,000 articles), sometimes referred to as mega journals, while half of the journals with at least one article have less than 192 articles in total. The semantic similarity metrics for recommendation would thus favour those mega journals, as they are more likely to contain similar articles due to their sheer size. The current algorithm takes this into account by limiting the number of top matches for the semantically close articles. This prevents the accumulation of a high number of articles with a very low score. Apart from the number of articles per journal, the number of articles per subject and the number of articles per language is also unevenly distributed. English dominates as a language with 77% of articles and medicine is the most common subject with 29.63% in contrast to Military Science with only 0.10%. Further research is needed to determine the exact impact of these imbalances on recommendation performance.

The general approach of suggesting journals based on semantic or bibliometric similarity does not cover all journal scopes. A journal focusing on a specific methodology but with a broader scope of topics will rank lower. Similarly, special issues of a journal can shift the topic for the recommendation algorithm as they contain a high concentration of articles on a certain (possibly niche) topic.

6 Conclusion

In this paper, we have presented a comprehensive experimental comparison of various retrieval methods for our open access journal recommendation system B!SON. The system combines semantic and bibliometric information to calculate a similarity score to the journals’ existing contents, and provides the user with a ranked list of candidate venues. The B!SON service is available online for use and testing.

The here-presented version represents a more advanced version of the prototype with respect to the contents of the original demo paper. Based on feedback from our user community, we improved the user interface and scoring functions. Other requests concern further improvements of user interface and usability. Features such as filter suggestions or automatic fetching of paper information from the pre-print server arXiv have been implemented and help with the tool’s usage. The community’s wishlist further includes more sophisticated methods for the exploration of the results, e.g. graph-based visualizations, an extension of the filtering options and an improved representation of the similarity

Table 3 Top@N accuracy for different methods with the “middle” requirements according to Table 1 applied to both training and test set

Recommendation method	Language model	Top@1	Top@5	Top@10	Top@15
Elasticsearch on title	–	0.084	0.174	0.214	0.234
Elasticsearch on abstract	–	0.147	0.303	0.377	0.420
Bibliometric search	–	0.189	0.312	0.342	0.353
Journal embeddings	BERT ^a	0.078	0.183	0.246	0.292
Journal embeddings	XML- RoBERTA	0.116	0.264	0.346	0.400
Journal embeddings	USE ^b	0.181	0.339	0.418	0.471
Article embeddings individual	XML- RoBERTA	0.067	0.170	0.238	0.285
Article embeddings top-100 combined	XML- RoBERTA	0.092	0.223	0.299	0.348
Article embeddings individual	USE	0.075	0.190	0.265	0.314
Article embeddings top-100 combined	USE	0.109	0.258	0.339	0.390
Article embedding + classifier	XML- RoBERTA	0.020	0.070	0.108	0.136
Article embedding + classifier	USE	0.140	0.309	0.404	0.462
Article embedding + BiLSTM + classifier	XML- RoBERTA	0.102	0.255	0.350	0.416

^aThe model “BERT- BASE- MULTILINGUAL” is abbreviated with “BERT” here

^bThe model “UNIVERSAL- SENTENCE- ENCODER- MULTILINGUAL- LARGE” is abbreviated with “USE” here

Table 4 Training time and testing time (in seconds) for the different methods for the experiment runs described in Table 3. There is not a specific training time for the bibliometric search included as it cannot be easily defined

Recommendation method	Language model	Training time	Test time
Elasticsearch on title	–	52	840
Elasticsearch on abstract	–	52	6062
Bibliometric search	–	–	8783
Journal embeddings	XML- RoBERTA	757	211
Journal embeddings	BERT ^a	1145	566
Journal embeddings	USE ^b	217	116
Article embeddings individual	XML- RoBERTA	414	712
Article embeddings top 100 combined	XML- RoBERTA	414	669
Article embeddings individual	USE	691	684
Article embeddings top 100 combined	USE	691	694
Article embedding + classifier	XML- RoBERTA	6878	919
Article embedding + classifier	USE	930	3420
Article embedding + BiLSTM + classifier	XML- RoBERTA	6915	1024

^aThe model “BERT- BASE- MULTILINGUAL” is abbreviated with “BERT” here

^bThe model “UNIVERSAL- SENTENCE- ENCODER- MULTILINGUAL- LARGE” is abbreviated with “USE” here

score. These action points have not been tackled yet, but we will explore possibilities in their respect in the future.

This paper reports on our experiments with embedding-based algorithms to improve the journal recommender. The structured comparative evaluation of the current recommendation methods shows promising results for the classifier built upon the pre-trained UNIVERSAL- SENTENCE- ENCODER model. We are currently working on integrating this technical solution into the productive recommendation system.

After our exploration into improving the semantic components of the recommendation, we now look into enhancing the bibliometric recommendation. Similar to the semantic methods, the bibliometric similarity computation shows a

tendency to favour larger journals. The more reference articles a journal contains, the more potential to find matching citations to the query set of references. The current implementation relies on a manually defined relevancy threshold which, in the future, should be computed automatically. Furthermore, we experiment with additional normalization methods, starting from established ones such as the Jaccard index [47] to balance between journals with a high or low publication output. The bibliometric component is computationally expensive. Adopting enhanced normalization in the productive system thus comes with challenges with respect to efficient implementation and computational resources.

Citation graph embeddings are another area which we are currently evaluating as they are commonly used for cita-

tion recommendation [48, 49]. The size of the COCI dataset makes this computationally expensive, however.

Beyond the scope of the B!SON project, it could be interesting to extend recommendations to other venues with open access options (e.g., conferences). Moreover, the integration of person-centred information, such as prior publication history and frequent co-authors, seems promising.

Acknowledgements B!SON is funded by the German Federal Ministry of Education and Research (BMBF) for a period of two years (funding reference: 16TOA034A) and partners with DOAJ and OpenCitations.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest Marco Tullney is a volunteer associate editor for the Directory of Open Access Journals (DOAJ). *TIB-Leibniz Information Centre for Science and Technology* and *Saxon State and University Library* are supporters of the Directory of Open Access Journals (DOAJ).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hartwig, J., Eppelin, A.: Which journal characteristics are crucial for scientists when selecting journals for their publications? Results tables of an online survey. Zenodo (2021). <https://doi.org/10.5281/zenodo.5728148>
- Entrup, E., Eppelin, A., Ewerth, R., Hartwig, J., Tullney, M., Wohlgemuth, M., Hoppe, A.: B!SON: A tool for open access journal recommendation. In: *Linking Theory and Practice of Digital Libraries*, pp. 357–364. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16802-4_33
- Pradhan, T., Pal, S.: A multi-level fusion based decision support system for academic collaborator recommendation. *Knowl.-Based Syst.* **197**, 105784 (2020). <https://doi.org/10.1016/j.knosys.2020.105784>
- Pradhan, T., Sahoo, S., Singh, U., Pal, S.: A proactive decision support system for reviewer recommendation in academia. *Expert Syst. Appl.* **169**, 114331 (2021). <https://doi.org/10.1016/j.eswa.2020.114331>
- Beel, J., Gipp, B., Langer, S., Breiteringer, C.: Research-paper recommender systems: a literature survey. *Int. J. Digit. Libr.* **17**(4), 305–338 (2016). <https://doi.org/10.1007/s00799-015-0156-0>
- Brack, A., Hoppe, A., Ewerth, R.: Citation recommendation for research papers via knowledge graphs. In: *International Conference on Theory and Practice of Digital Libraries*, TPD 2021, Virtual Event, September 13–17, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12866, pp. 165–174. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86324-1_20
- Schäfermeier, B., Stumme, G., Hanika, T.: Towards Explainable Scientific Venue Recommendations. *arXiv:2109.11343* [cs] (2021)
- ZhengWei, H., JinTao, M., YanNi, Y., Jin, H., Ye, T.: Recommendation method for academic journal submission based on doc2vec and XGBoost. *Scientometrics* **127**(5), 2381–2394 (2022). <https://doi.org/10.1007/s11192-022-04354-1>
- Nguyen, D., Huynh, S., Huynh, P., Dinh, C.V., Nguyen, B.T.: S2CFT: A new approach for paper submission recommendation. In: *SOFSEM 2021: Theory and Practice of Computer Science* vol. 12607, pp. 563–573. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67731-2_41
- Rollins, J., McCusker, M., Carlson, J., Stroll, J.: Manuscript Matcher: A Content and Bibliometrics-based Scholarly Journal Recommendation System. In: *Proceedings of the Fifth Workshop on Bibliometric-enhanced Information Retrieval (BIR) Co-located with the 39th European Conference on Information Retrieval (ECIR 2017)*, Aberdeen, UK, April 9th, 2017, (2017). <http://ceur-ws.org/Vol-1823/paper2.pdf>
- Ghosal, T., Chakraborty, A., Sonam, R., Ekbal, A., Saha, S., Bhattacharyya, P.: Incorporating full text and bibliographic features to improve scholarly journal recommendation. In: *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 374–375. IEEE, Champaign, IL, USA (2019). <https://doi.org/10.1109/JCDL.2019.00077>
- Zawali, A., Boukhris, I.: Academic venue recommendation based on refined cross domain. In: *Intelligent Systems Design and Applications* vol. 418, pp. 1188–1197. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-96308-8_110
- Pradhan, T., Pal, S.: A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity. *Futur. Gener. Comput. Syst.* **110**, 1139–1166 (2020). <https://doi.org/10.1016/j.future.2019.11.017>
- Yu, S., Liu, J., Yang, Z., Chen, Z., Jiang, H., Tolba, A., Xia, F.: PAVE: personalized academic venue recommendation exploiting co-publication networks. *J. Netw. Comput. Appl.* **104**, 38–47 (2018). <https://doi.org/10.1016/j.jnca.2017.12.004>
- Vara, N., Mirzabeigi, M., Sotudeh, H., Fakhrahmad, S.M.: Application of k-means clustering algorithm to improve effectiveness of the results recommended by journal recommender system. *Scientometrics* (2022). <https://doi.org/10.1007/s11192-022-04397-4>
- Schäfermeier, B., Stumme, G., Hanika, T.: Towards explainable scientific venue recommendations. *arXiv:2109.11343* [cs] (2021)
- Abbasi, I.I., Abbas, M.A., Hammad, S., Jilani, M.T., Ahmed, S., Nisa, S.U.: A hybrid approach for the recommendation of scholarly journals. In: *2020 International Conference on Information Science and Communication Technology (ICISCT)*, pp. 1–6. IEEE, KARACHI, Pakistan (2020). <https://doi.org/10.1109/ICISCT49550.2020.9080032>
- Cuong, D.V., Nguyen, D.H., Huynh, S., Huynh, P., Gurrin, C., Dao, M.-S., Dang-Nguyen, D.-T., Nguyen, B.T.: A framework for paper submission recommendation system. In: *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 393–396. ACM, Dublin Ireland (2020). <https://doi.org/10.1145/3372278.3391929>
- Wang, D., Liang, Y., Xu, D., Feng, X., Guan, R.: A content-based recommender system for computer science publications. *Knowl.-Based Syst.* **157**, 1–9 (2018). <https://doi.org/10.1016/j.knosys.2018.05.001>
- Huynh, S.T., Huynh, P.T., Nguyen, D.H., Cuong, D.V., Nguyen, B.T.: S2RSCS: An efficient scientific submission recommendation system for computer science. In: *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices* vol.

- 12144, pp. 186–198. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55789-8_17
21. Feng, X., Zhang, H., Ren, Y., Shang, P., Zhu, Y., Liang, Y., Guan, R., Xu, D.: The deep learning-based recommender system "Pubmender" for choosing a biomedical publication venue: development and validation study. *J. Med. Internet Res.* **21**(5), 12957 (2019). <https://doi.org/10.2196/12957>
22. Kobs, K., Koopmann, T., Zehe, A., Fernes, D., Krop, P., Hotho, A.: Where to submit? Helping researchers to choose the right venue. In: Findings of the association for computational linguistics: EMNLP 2020, pp. 878–883. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.78>
23. Son, H.T., Tan Phong, H., Dac, N.H.: An efficient approach for paper submission recommendation. In: 2020 IEEE Region 10 Conference (TENCON), pp. 726–731. IEEE, Osaka, Japan (2020). <https://doi.org/10.1109/TENCON50793.2020.9293909>
24. Kang, N., Doornenbal, M.A., Schijvenaars, R.J.A.: Elsevier journal finder: recommending journals for your paper. In: Proceedings of the 9th ACM Conference on Recommender Systems. RecSys '15, pp. 261–264. Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2792838.2799663>
25. Schuemie, M.J., Kors, J.A.: Jane: suggesting journals, finding experts. *Bioinformatics (Oxford, England)* **24**(5), 727–728 (2008). <https://doi.org/10.1093/bioinformatics/btn006>
26. Gündoğan, E., Kaya, M., Daud, A.: Deep learning for journal recommendation system of research papers. *Scientometrics* (2022). <https://doi.org/10.1007/s11192-022-04535-y>
27. ZhengWei, H., JinTao, M., YanNi, Y., Jin, H., Ye, T.: Recommendation method for academic journal submission based on doc2vec and XGBoost. *Scientometrics* **127**(5), 2381–2394 (2022). <https://doi.org/10.1007/s11192-022-04354-1>
28. Medvet, E., Bartoli, A., Piccinin, G.: Publication venue recommendation based on paper abstract. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, pp. 1004–1010. IEEE, Limassol, Cyprus (2014). <https://doi.org/10.1109/ICTAI.2014.152>
29. Ghosal, T., Chakraborty, A., Sonam, R., Ekbal, A., Saha, S., Bhattacharyya, P.: Incorporating full text and bibliographic features to improve scholarly journal recommendation. In: 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp. 374–375. IEEE, Champaign, IL, USA (2019). <https://doi.org/10.1109/JCDL.2019.00077>
30. Huynh, S.T., Dang, N., Nguyen, D.H., Huynh, P.T., Nguyen, B.T.: FPSRS: A fusion approach for paper submission recommendation system. *arXiv:2205.05965 [cs]* (2022)
31. Ghosal, T., Raj, A., Ekbal, A., Saha, S., Bhattacharyya, P.: A deep multimodal investigation to determine the appropriateness of scholarly submissions. In: 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp. 227–236. IEEE, Champaign, IL, USA (2019). <https://doi.org/10.1109/JCDL.2019.00039>
32. Eaton, M.: The last days of the Open Journal Matcher. <https://kingsboroughlibtech.commons.gc.cuny.edu/2022/07/29/the-last-days-of-the-open-journal-matcher/> Accessed 2022-11-09
33. Hartwig, J., Eppelin, A.: Which journal characteristics are crucial for scientists when selecting journals for their publications? Results Tables of an Online Survey. <https://doi.org/10.5281/zenodo.5728148>
34. Hicks, D., Wouters, P., Waltman, L., de Rijcke, S., Rafols, I.: Bibliometrics: The Leiden Manifesto for research metrics. *Nature* **520**(7548), 429–431 (2015). <https://doi.org/10.1038/520429a>
35. Directory of Open Access Journals. <https://doaj.org/> Accessed 2022-11-09
36. OpenCitations: COCI CSV dataset of all the citation data. figshare (2022). <https://doi.org/10.6084/m9.figshare.6741422.v18>
37. Martín-Martín, A., Thelwall, M., Orduna-Malea, E., Delgado López-Cózar, E.: Google scholar, microsoft academic, scopus, dimensions, web of science, and opencitations'coci: a multi-disciplinary comparison of coverage via citations. *Scientometrics* **126**(1), 871–906 (2021). <https://doi.org/10.1007/s11192-020-03690-4>
38. Nishioka, C., Färber, M.: Evaluating the availability of open citation data. In: Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) Co-located with the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), Paris, France, July 25, 2019. CEUR Workshop Proceedings, vol. 2414, pp. 123–129. CEUR-WS.org, Germany (2019)
39. Hannah Hope: Unboxing the Journal Checker Tool | Plan S. <https://www.coalition-s.org/blog/unboxing-the-journal-checker-tool/> Accessed 2022-06-01
40. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.* **3**(4), 333–389 (2009). <https://doi.org/10.1561/15000000019>
41. Kessler, M.M.: Bibliographic coupling between scientific papers. *Am. Doc.* **14**(1), 10–25 (1963). <https://doi.org/10.1002/asi.5090140103>
42. Quang, M.N., Rogers, T., Hofman, J., Lanham, A.B.: New framework for automated article selection applied to a literature review of Enhanced Biological Phosphorus Removal. *PLoS ONE* **14**(5), 0216126 (2019). <https://doi.org/10.1371/journal.pone.0216126>
43. Quang, M.N., Rogers, T., Hofman, J., Lanham, A.B.: Difference between bibliographic coupling and co-citation. *PLoS ONE* (2019). <https://doi.org/10.1371/journal.pone.0216126.g001>
44. Beltagy, I., Lo, K., Cohan, A.: SciBERT: A pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, pp. 3613–3618. Association for Computational Linguistics, New York, NY, USA (2019). <https://doi.org/10.18653/v1/D19-1371>
45. Gündoğan, E., Kaya, M., Daud, A.: Deep learning for journal recommendation system of research papers. *Scientometrics*, 1–21 (2022). <https://doi.org/10.1007/s11192-022-04535-y>
46. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. *arXiv* (2019). <https://doi.org/10.48550/ARXIV.1911.02116>
47. Leydesdorff, L.: On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index. *J. Assoc. Inf. Sci. Technol.* **59**(1), 77–85 (2008). <https://doi.org/10.1002/asi.20732>
48. Pornprasit, C., Liu, X., Kertkeidkachorn, N., Kim, K.-S., Noraset, T., Tuarob, S.: Convcn: A cnn-based citation network embedding algorithm towards citation recommendation. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020. JCDL '20, pp. 433–436. Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3383583.3398609>
49. Ali, Z., Qi, G., Muhammad, K., Bhattacharyya, S., Ullah, I., Abro, W.: Citation recommendation employing heterogeneous bibliographic network embedding. *Neural Computing and Applications*, 1–14 (2021). <https://doi.org/10.1007/s00521-021-06135-y>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.