A Hybrid Intelligent Approach Combining Machine Learning and a Knowledge Graph to Support Academic Journal Publishers Addressing the Reviewer Assignment Problem (RAP)

Dietrich Rordorf[1,2], Josua Käser[1], Alfredo Crego[1] and Emanuele Laurenzi[1,†]

[1] *School of Business, University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, CH-4600 Olten, Switzerland*
[2] *MDPI AG, St. Alban-Anlage 66, 4052 Basel, Switzerland*

Abstract
This paper presents a hybrid intelligent approach that combines natural language processing (NLP) and knowledge engineering to address the Reviewer Assignment Problem (RAP) in scientific peer-review. The approach uses NLP techniques to match a new document with subject experts, and it employs a knowledge graph to identify conflicts of interest (COIs) between the authors of a document and potential reviewers. The approach detects three types of COIs: direct co-authorship, second-level co-authorship, and collaborators from the same institutions. Further, it uses semantic text similarity (STS) matching for peer-reviewing of documents in journals, where potential reviewers are screened from large literature databases. The research approach follows the Design Science Research methodology, where a prototypical system is designed based on the requirements elicited from both the literature and from primary data collection conducted in a publishing house. The approach is evaluated by implementing real-world use cases in the working prototype and by conducting a focus group with potential users, i.e., editors.

Keywords
decision support system, reviewer assignment problem (RAP), conflicts of interest (COIs), semantic text similarity (STS), vector searchs, graph database

1. Introduction

Peer-review refers to the evaluation of scientific articles by one or more subject experts that have similar competencies as the authors of the scientific articles and is used to safeguard the quality of research publications [1]. A high-quality peer-review process is ensured by assigning adequate experts as the reviewers of a scientific article. Within this process, one has to address the Reviewer Assignment Problem (RAP) [2, 3] to avoid biased outcomes [4]. As part

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.
[†] Main contact.
dietrichhanspaul.rordorf@students.fhnw.ch (D. Rordorf); josua.kaeser@students.fhnw.ch (J. Käser); alfredoetienne.cregocorot@students.fhnw.ch (A. Crego); emanuele.laurenzi@fhnw.ch (E. Laurenzi)

of the RAP, the identification of conflicts of interest (COIs) between authors and reviewers is one of the most crucial challenging, and time-consuming activities as it is commonly done semi-automatically [5]. This is especially problematic for large academic journal publishers such as Elsevier or MDPI that have a substantial number of papers and reviewers involved. The complexity lies behind the activities of checking for COIs that go beyond the direct co-authorship between any of the co-authors with the potential reviewers, or via collaborators at co-authors' institutions. Recently some approaches have started to consider the academic social network and collaboration distance between authors and reviewers as a criterion for a fair and balanced reviewer assignment [6, 7]. These solutions are geared towards automating the reviewer assignment optimization problem in the setting of peer review in conferences, where a number of documents is assigned to a fixed number of peer reviewers. Our work addresses a different class of complexity where the identification of COIs is done on a large scale, among millions of documents and authors.

In this paper, we present a decision support system that, given a journal article, recommends potential reviewers and transparently shows their COIs for final human decision-making. The system is designed considering scale as an important requirement. The system is split into two parts. Firstly, a natural language processing (NLP) component that aims to match a new document with subject experts. The component allows for semantic text similarity (STS) matching using document embeddings, which are created through the SPECTER language model [8]. Secondly, a knowledge graph component allows identifying COIs by navigating the graph. Our approach considers three types of COIs: (a) direct co-authorship, (b) second-level co-authorship, and (c) current and former collaborators of the same institute(s).

The paper is structured as follows. In Section 2 the related work is discussed, which comprises approaches in automating reviewer assignments, paper-reviewer match-making, STS matching at scale, reviewer assignment algorithms, and resolution of COIs. We conclude the section by pointing to promising approaches for the resolution of COIs, which involve solely knowledge graphs and hybrid artificial intelligent approaches. Next, the methodology is described in Section 3. Then, Section 4 elaborates on the tackled challenge, its analysis, and derivation of design requirements. The solution design is described in Section 5. The proof of concept is presented in Section 6 and comprises (a) the implementation of the approach in a working prototype, (b) evaluation of the approach with respect to a use case, and in Subsection 6.1, (c) the second evaluation that focuses on the perceived usefulness and usability of the prototype. Finally, the conclusion and outlook are discussed in 7.

2. Related Work

In this chapter, we first provide an overview of existing approaches in the match-making of reviewers with scholarly papers and the reviewer assignment problem (RAP). We then elaborate on knowledge graphs and their application to the detection of COIs. Finally, we point out hybrid intelligent approaches, which combine machine learning and knowledge engineering for the resolution of COIs.

2.1. Approaches in Automating Reviewer Assignments

The reviewer assignment typically depends on the domain knowledge of a single meta-reviewer or a panel of meta-reviewers that coordinate the process of selecting and inviting suitable reviewers for each document [9]. Apart from matching the topic of the document to the topical expertise of reviewers, other factors such as COIs or bidding preferences are considered [3].

Zhao & Zhang reviewed a number of systems for automating reviewer matching in RAP before [10]. They propose to characterize reviewer assignment systems as three-staged systems: Firstly, the construction of a reviewer database. Secondly, the match-making of papers with potential reviewers by computing similarity scores between the paper and previous publications of potential reviewers. Thirdly, a reviewer assignment optimization algorithm [10]. The step of constructing a reviewer database can also be replaced with a database of past publications.

2.1.1. Paper-Reviewer Match-Making

The paper-reviewer match-making involves the determination of the degree of compatibility between a paper and each potential reviewer [11]. Several Natural Language Processing (NLP) techniques have been proposed to determine this compatibility. These techniques leverage word frequency information, topic information, or deep semantic information.

In the domain of word frequencies, the most commonly used algorithm is the Term Frequency-Inverse Document Frequency (TF-IDF), which determines the relevance of words in the manuscript and reviewers' profiles [12, 13]. The cosine similarity metric is then used to calculate the degree of match between the manuscript and potential reviewers. The Toronto Paper Matching System (TPMS) [14] is a famous example of such a system using text matching scores at its core and is extensively used in computer science conferences. In the domain of topic information, Mimno & McCallum leveraged the technique of Latent Dirichlet Allocation (LDA) to transform the manuscripts of reviewers and authors into topic features [15].

However, techniques based on word frequencies and topic modelling are being superseded with deep neural network models. Kotak *et al.* developed an evaluation framework and found that reviewer recommender systems using Contextual Neural Topic Modelling (CNTM, using word embeddings) and Sentence-BERT (SBERT, using sentence embeddings) were superior to other techniques [11]. This is likely due to the word- and sentence-level contextual understanding employed in these models [11].

More recently transformer-based language models trained specifically on scholarly documents emerged that are suitable to transform and represent scholarly documents as vector embeddings, including Scibert [16] and SPECTER [8]. In the case of SPECTER, vector embeddings are representations of the features of the documents including the academic citation graph, and can be used for downstream tasks without task-specific fine-tuning [8]. One such downstream task is STS matching by e.g., computing the Euclidean distance or cosine similarity between two vectors.

2.1.2. Semantic Text Similarity Matching at Scale

The k-NN search is computationally expensive. Computing the cosine similarity of vector embeddings to find the most similar vectors may work well for small conferences with a limited number of contributions and reviewers. In this specific setting, similarity scores need to be computed for a finite amount of paper pairs to find the nearest neighbors. In the setting of scholarly journals, many journal editors may not want to limit themselves to a fixed-size review panel (e.g., the editorial board), but will conduct extensive literature research in specialized scholarly literature databases to identify potential reviewers from previous publications. Such literature databases contain millions of documents and computing cosine or Euclidian distances of vector embeddings at run-time is too slow. Solutions such as Faiss or Weaviate allow the creation of an index of a k-NN graph for large collections of vectors, which can be searched via approximate nearest neighbors (ANN) algorithms [17]. A commonly used ANN algorithm is the Hierarchical Navigable Small World (HNSW), which performs well in the ANN benchmark (high query throughput, high recall) [18].

2.1.3. Reviewer Assignment Algorithms

The reviewer assignment systems' third stage entails algorithms for the optimal selection and paper distribution to reviewers [10]. According to Long *et al.* and cited in Zhao & Zhang, the algorithms can be divided into two distinct groups based on their information retrieval or matching-based approach [19, 10]. Retrieval-based approaches are used in one-to-several reviewer assignment scenarios, where each paper is matched against the reviewer database one at a time [10]. This is typically the case for journals, where editors want to choose the most appropriate reviewers from past publications or a large pool of potential reviewers. The matching-based approach is used in a many-to-many reviewer assignment scenario, where a batch of papers is assigned to a fixed pool of reviewers [10]. This is typically the case for conferences. According to Shah, such reviewer assignment optimization algorithms typically maximize an overall sum of scores under consideration of other constraints, such as the number of documents per reviewer [20]. For both types of algorithms, identifying the most suitable reviewer for a paper also entails avoiding cooperative or competitive COIs between the reviewer and the author as part of such constraints [19].

2.1.4. Resolution of Conflicts of Interest

Recently some approaches to RAP have started to consider the academic social network and collaboration distance between authors and reviewers as a criterion for a fair and balanced reviewer assignment [6, 7]. Li *et al.* created a new score that combines the topic similarity and the collaboration distance between an article's authors and potential reviewers as a single metric to maximize [6]. Using a single score for ranking, their approach aims at automating the reviewer assignment optimization in the setting of conferences, where a fixed number of documents is typically distributed to a fixed number of reviewers. Nugroho *et al.* split the assignment into two scores. Firstly, they compute the topic similarity via Latent Dirichlet Allocation (LDA). Secondly, they compute the cosine similarity between vector representations of the author node and the reviewer node. Both scores are then combined to create a ranking of the recommended reviewers [7]. Their approach shows the potential to provide more control to the editor, as the two types of scoring could be split and treated sequentially. However, in both approaches the score (based on collaboration distance or the similarity of graph embeddings) allow for little insights into the types of COIs that affect the reviewer candidate.

2.2. Knowledge Graphs for the Resolution of Conflicts of Interest

Knowledge graphs serve as an essential component for the development of advanced search and recommender systems [21]. The development of recommender systems, which rely on traditional semantic similarity matching and graph embedding-based recommendation, offers promising application prospects. Such approaches, when used in conjunction, complement each other and enhance the efficacy of the recommender system [22]. A graph database has the potential to increase the efficiency of the process to screen for potential referees by exploiting background information present in the graph. A graph database is especially suited to screen for COIs between the author and potential reviewers by navigating the graph between the author's node and the potential reviewer's node. Structured relationships, denoted by subject-predicate-object, enable the establishment of meaningful links between entities in a graph. This approach facilitates not only the identification of direct COIs among co-authors, but also second-level COIs among co-authors of co-authors. Thus, the technique provides a comprehensive screening mechanism for detecting potential COIs in scientific collaborations [23].

2.3. Hybrid Artificial Intelligent Approaches for the resolution of Conflicts of Interest

Several hybrid AI approaches have been researched to combine knowledge graphs and computational learning. In the RDF2Vec approach [24] each node of a Resource Description Framework (RDF) graph is converted into a numeric vector which is then used to build a neural language model. The creation of a vector space embedding for large RDF graphs can be computationally challenging, although this step has to be performed only once, as the embeddings can be reused for different tasks. Ristoski *et al.* show a superior performance of the RDF2Vec approach for document similarity or recommender systems [24]. In our approach we are using the embeddings of SPECTER which already includes the citation graph of scientific papers [8], therefore partly covering information that we would be able to include in a neural language model using RDF2Vec. Moreover, we aim for a hybrid intelligent approach, where the hybrid AI and editors collaborate.

3. Methodology

In this chapter, we provide an overview of the research methodology. The research design follows the Design Science Research (DSR) methodology [25].

In the problem awareness phase, we combine secondary data with primary data for the derivation of a set of design requirements. As secondary data, we conducted a qualitative literature review on peer-review, COIs detection, expert recommender systems, and hybrid intelligent systems. As primary data, we conducted semi-structured interviews with a professional managing editor from the academic journal publisher MDPI[1] to gather insights into a current peer-review process at a large journal. The interview lasted one hour and was recorded with the permission of the interviewee.

In the suggestion phase, we elaborate the design for a novel decision support approach by addressing the requirements and problems identified during the problem awareness phase.

In the development phase, the approach is implemented into a running prototypical system, which includes a backend and a front-end application. The backend includes a vector search index with document embeddings and an RDF(S) graph, which is derived from real-world data of MDPI publications from 2020-2021. The frontend application is a user interface for editors.

In the evaluation phase, our approach is two-fold. Firstly, a real-world use case is implemented in the prototypical system to prove the correctness of the artifact. Secondly, a focus group with editors is used to prove the usefulness of the prototypical system via its front-end user interface.

The below sections elaborate on each of the DSR phases.

4. The Editorial Process Challenge

In this section, we first present our findings relating to the current editorial process to identify and screen peer-reviewers. We take the example of peer-review for a typical MDPI journal. Based on the process findings and the literature review from Chapter 2, we then established a set of requirements for the design of the novel approach to support editors in finding appropriate potential reviewers and resolving their potential COIs. The solution design is derived from the requirements and implemented as a prototypical software system.

4.1. As-is Editorial Process of a Manuscript

In this sub-section, we present the current editorial process of a manuscript (i.e., journal) and the problems that may arise in the identification of potential reviewers. Findings were derived by interviewing a professional managing editor and by analyzing MDPI training material, which is used to train in-house editorial staff.

The following description of the editorial process is underpinned by the graphical process depicted in Figure 1. Each manuscript is screened by the editorial staff before it is sent out for peer-review. Typically, authors are asked to provide a positive and a negative list of reviewers (i.e. author's bidding). A positive list of reviewers includes persons whom the authors of the manuscript think are qualified to conduct an independent review of the manuscript. The negative list includes persons whom the authors do not want to be contacted for the peer-review of their manuscript. This typically includes past collaborators (i.e. self declared COIs) or competing research groups. Besides names provided by authors, the editorial staff will conduct a literature review to identify authors that have published in the same domain and may thus potentially qualify as reviewers. All potential reviewers are screened for possible COIs, qualifications, and past track records.

[1] https://www.mdpi.com/

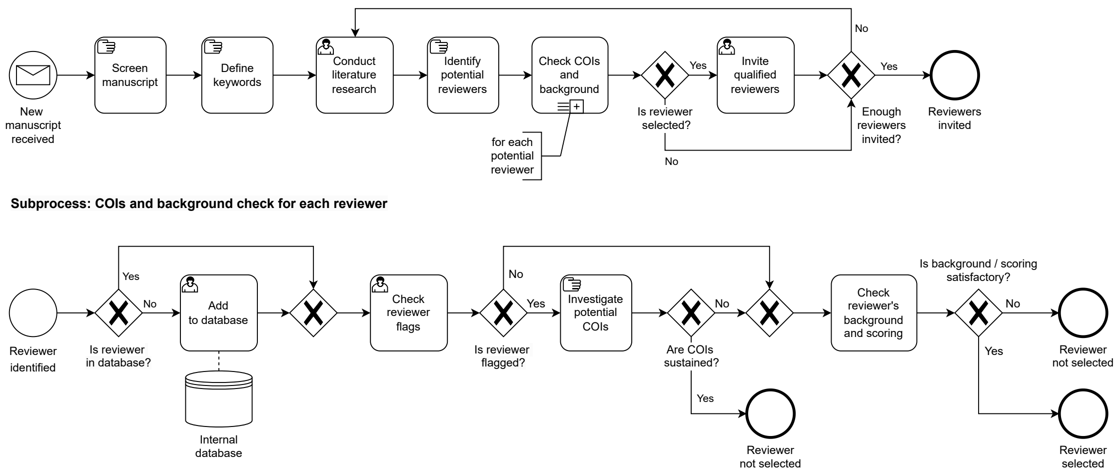

Figure 1: Current editorial process to identify and screen potential reviewers (in BPMN 2.0).

To conduct the literature search, the editor identifies keywords from the title and abstract of the manuscript. Typically the editor tries to summarize the keywords into more general concepts and find synonymous keywords to enlarge the pool of related search results. The step of summarizing keywords into more general concepts requires high expertise in the domain of the manuscript. Less experienced editors or editors with differing backgrounds (e.g. with more editorial skills rather than scientific) are usually not able to apply this search strategy proficiently and may have to deal with an inadequate or limited number of search results.

The identified keywords are then used against specialized literature databases such as the Web of Science and Google Scholar to find literature from recent years. Authors of such recent related literature are considered potential reviewers for the manuscript at hand. Potential reviewer names are added to the manuscript processing system. The manuscript processing system will automatically flag potentially problematic reviewers. There are several flags indicating different types of potential problems:

1. COIs:
    - Co-authorship: one or more authors and the potential reviewer may have published together in past 5 years. This check is performed based on matching authors' names and the reviewer's names (which may pose additional disambiguation problems) in authorship lists from the past 5 years as indexed in Scilit.
    - Collaborators from the same institute: authors and potential reviewers may be working at the same institute. This check is performed by comparing the hostname of the authors' and reviewers' email addresses.
2. Publication ethics: the reviewer is listed in previous publication ethics cases, such as citation cartels (asking authors to add citations to their own work), fake or poor quality reviews, plagiarism, etc.

3. Opt-out: the reviewer opted out of performing peer-reviews for a particular journal or for the publisher.

In the case of co-authorship COIs and publication ethics, the editorial staff proceeds with the manual verification of the reported flag. For co-authorship COIs, this entails a laborious manual verification of the purportedly shared publication: the editorial staff verifies that the authors and the potential reviewer in listed publications are the same person. In some cases the initial flag set by the system can not be substantiated, e.g., another person sharing the same name as the reviewer or the author. In this case, the editor can proceed in the process with this potential reviewer. The interviewee noted that currently, the system is not capable of identifying a second-level co-authorship: a co-authorship between a co-author of one of the manuscript's authors and the potential reviewer. Further, he noted that a manual check for second-level authorship is complex and would involve several recursive literature searches. Thus, editors currently do not verify for second-level co-authorship. Further, the system sometimes misses flagging collaborators from the same institute due to different email address endings, the absence of email addresses on the author or reviewer profile, or due to secondary affiliations (a reviewer may have a side-affiliation with one of the co-authors' institute but use the email address from his main affiliation).

Once the flags set by the system have been resolved (or no flags reported by the system), the editors proceed to a further background check of the reviewer. This includes looking-up previous peer-review performance in the internal system. The system includes such details as the past delivery time for review reports and the quality of the delivered review report. The reviewer may have for instance repeatedly provided very superficial comments or "template reports" in the past and may hence not be a good choice. Additionally, the reviewers have to meet the following criteria to be considered qualified as a reviewer:

C1 Academic qualification: only scholars having a PhD or equivalent degree are considered potential reviewers. This check excludes graduates- and PhD-students from being considered as reviewers.

C2 Expertise: several publications in the field over the past five years, preferably as the lead author and in international journals.

C3 Citation record: above average h-index or i10-index compared to typical values in the field.

The process is subject to some variations depending on the journal: some of the journal's academic editors may want to be involved in screening reviewers, or approving reviewers screened by the journals' editorial staff before the invitation. In a few cases, the academic editor may want to screen the reviewers by themselves. The figure shows the typical process variant handled by a single in-house or academic editor.

4.2. Requirements

Based on the identified process, the interview with the editor, and contextual information of the publisher MDPI, the following requirements for the solution design were defined:

R1 The approach should match related publications via semantic text similarity (STS).
Rationale: remove the need for editors to extract keywords from titles and abstracts of manuscripts. Further, the solution should be accessible to editors that are not experts in the domain of the manuscript.

R2 The STS matching and COI resolution should scale to millions of documents.
Rationale: support editors at large journals which do not have a finite list of reviewers from which to choose. Instead, journal editors conduct a literature search to screen for potential reviewers from related literature.

R3 The approach should include a database of past publications with disambiguate entities (authors, institutes).
Rationale: in order to load the author network into a meaningful graph database, author and institution entities need proper disambiguation.

R4 The approach should ease the process of checking for COIs.
Rationale: the approach should remove the need for laboriously checking COIs reported based on name matching by, e.g., using disambiguate author entities.

R5 The approach should introduce the checking of second-level authorship.
Rationale: this type of COI is not resolved at all today due to the amount of manual work and the need for recursive literature research to solve this task.

R6 The approach should provide reasoning for proposing or excluding a reviewer.
Rationale: in case of doubts, the editors should have the possibility to trace back the proposed reviewer name to a publication to make a final decision. Also, the editor should be able to verify a COI reported by the approach.

R7 The approach should follow a hybrid intelligent approach.
Rationale: the approach should support the editors in making decisions while allowing the editor to fine-tune settings and engage with the search results.

5. The Proposed Solution Design

In this section, we present our solution design based on the requirements derived from the analysis of the editorial process challenge. We scoped our solution toward two major problems. Firstly, we automate the matching of previous related literature to remove limitations introduced by the classical keyword-based information retrieval approach (*R1*, *R2*). Secondly, our solution improves COI screening by introducing a directed authorship graph allowing for direct and indirect co-authorship screening, as well as improved collaborator screening via affiliations and side-affiliations (*R3*–*R5*). Hence, the proposed solution consists of a two-step approach. The first step focuses on employing machine learning. It matches the manuscript with past publications through STS to build a pool of potential reviewers. The second step focuses on machine reasoning. It resolves COIs of each potential reviewer with all co-authors via the academic authorship graph. The proposed solution is complemented by a user interface for the editors, and an ETL pipeline to load and transform existing publication data into the solution. Figure 2 depicts the solution.

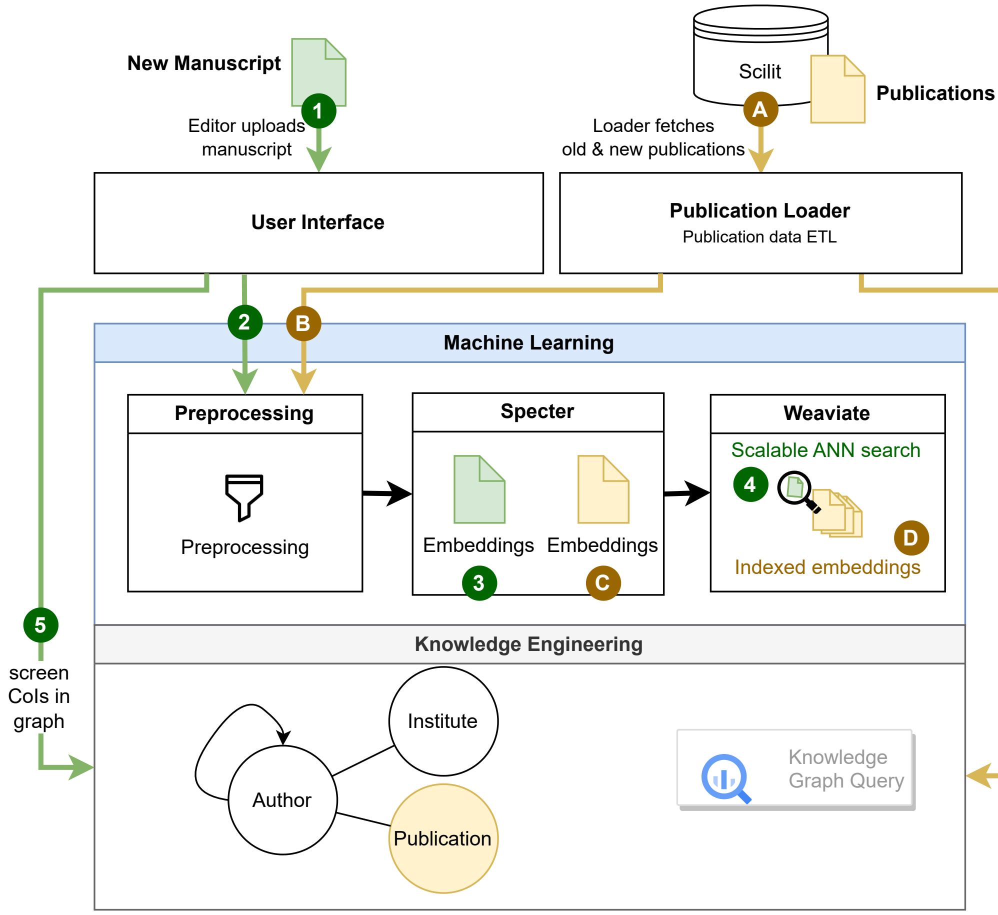

Figure 2: Overview of the proposed hybrid intelligent solution design combining Machine Learning and Knowledge Engineering.

5.1. Semantic Text Similarity (STS)

To simplify the process and to address *R1*, we propose to use an NLP-based approach in matching related articles through STS. Transformer-based language models are ideal for this task: they are capable of representing a document as a vector (document embeddings). The BERT-based transformer SPECTER developed by AllenAI is a good candidate due to the nature of the scholarly documents used in its training and its ability for performing downstream tasks without fine-tuning [8]. Using document embeddings also addresses one of the problems we identified in the current process, namely the difficulty in keyword-based search strategy faced by editors that are not very familiar with a manuscript's topic.

5.2. Vector Index and ANN Search

To support the use case at journals as described in *R2* the solution should support holding millions of previous publications. According to Scilit, there are 35 million scholarly documents published over the past 5 years[2]. In this setting, the STS matching can only be performed if document embeddings are pre-computed and stored in an index. After evaluation of the documentation of two systems – Faiss [17] and Weaviate [26] – we propose to use Weaviate as the vector index and search engine due to the in-built horizontal scalability and support for ANN via the HNSW algorithm. Additionally, Weaviate supports storing and searching for additional properties, such as titles, abstracts, authors' countries or the publication outlet. This allows for more flexible search queries if the system is to be expanded in the future in support of *R7*. An example could be to allow the user to limit the vector search to a specific publication outlet (e.g., only search for related vectors within the *Lancet* journal), year range, or publications authored by persons in certain countries.

5.3. Resolving COIs via Graph Database

We use the RDF graph database GraphDB[3] to resolve COIs between authors and potential reviewers. To address the requirement of scalability (*R2*) we construct SPARQL queries to extract sub-graphs for each type of conflict (direct authorship, second-level authorship, and institute collaborators) and run each query for each manuscript co-author, thereby addressing *R4* & *R5*. We then extract the author IDs that appear in each sub-graph and store them in a data frame by type of conflict. Finally, we scan the data frame for author IDs that also appear in the pool of potential reviewers and flag the potential reviewers according to the type of COI that was identified. The flagged potential reviewers and the type of COI that was found are shown in the user interface, thereby addressing *R6*.

5.4. Publication Data ETL

We leverage on the Scilit database from MDPI, which contains past publications and disambiguate author and institute data and addresses *R3*. Authors of semantically related papers are considered as potential reviewers, thereby eliminating the need for creating a separate reviewers database. An ETL pipeline transforms publication data twofold. Firstly, it is used to create document embeddings with SPECTER and stored into the vector index. Secondly, the publication data is transformed into an RDF graph and loaded into the graph database.

6. Proof of Concept

The proposed solution design has been instantiated as a prototypical software system. The system is composed by the following four components:

- Vector Search Engine: is an index of vectors in Weaviate representing the document embeddings of past scholarly publications created with SPECTER. The component

[2]see: https://app.scilit.net/publications and apply the "Past 5 years" filter from the left-side menu.
[3]https://graphdb.ontotext.com/

supports ANN search through the in-built implementation of the Hierarchical Navigable Small World (HNSW) algorithm via a REST API endpoint.

- Graph Database: graph representation of past publications in GraphDB, including the co-authorship network and institutional affiliations. The component supports the resolution of COIs between potential reviewers and authors (direct co-authorship, second-level co-authorship, and past and present collaborators at the same institutes) by querying via the SPARQL endpoint.

- Backend Application: backend application in Python offering Flask API endpoints to access the business logic, compute document embeddings with SPECTER, and access data from the other layers (graph database, vector search). This application also includes the ETL pipeline to load data from Scilit, convert it into an RDF graph and batch-import into GraphDB via Turtle files.

- Frontend Application: frontend with a graphical user interface for journal editors built in Nuxt.js and Vue.js.

To evaluate the approach, we implemented a use case with real-world data. Specifically, publication data from MDPI for 2020-2021 was obtained from Scilit, transformed into RDF and loaded into GraphDB. The data set consists of 400,000 publications, 1,200,000 unique authors, and 22,500 institutes. Additionally, the title and abstracts were concatenated and document embeddings computed in the Python backend application. Embeddings were batch-pushed to Weaviate via its REST API. The ETL processing of this data set took a total of 16 hours. The application, GraphDB and Weaviate were deployed on a virtual machine (VM) of type e2-standard-2 with 8 GiB memory and 80 GiB disk size on the Google Cloud. During the ETL import, we temporarily switched the VM to the e2-highmem-2 type with 16 GiB memory.

The frontend user interface is divided into an introductory landing page and three subsequent simple process steps:

1. In the first step, the user can type or copy the title, abstract and authors of the new manuscript. A random sample generator button provides example data for a quick demonstration, but the users can enter any manuscript they would like to evaluate (Figure 3). To uniquely identify manuscript authors, the user is asked to enter the email addresses of the co-authors. The email addresses are not saved and are only used to map the authors to the unique author ID in Scilit, which is subsequently needed to query the graph for COIs. For our evaluation, we focused on the manuscript titled "Buckwheat in Tissue Culture Research: Current Status and Future Perspectives" [4].

2. In the second step, a list of the most matching search results is presented with all publications matching the manuscript, sorted by descending score. The score is provided by the Weaviate vector search engine and represents an inverted and normalized angular cosine distance in the range 0.0-1.0. The editor can select the publications that are most relevant to the topic. By default, the solution shows the 25 most matching publications (Figure 4).

[4]https://www.mdpi.com/1422-0067/23/4/2298

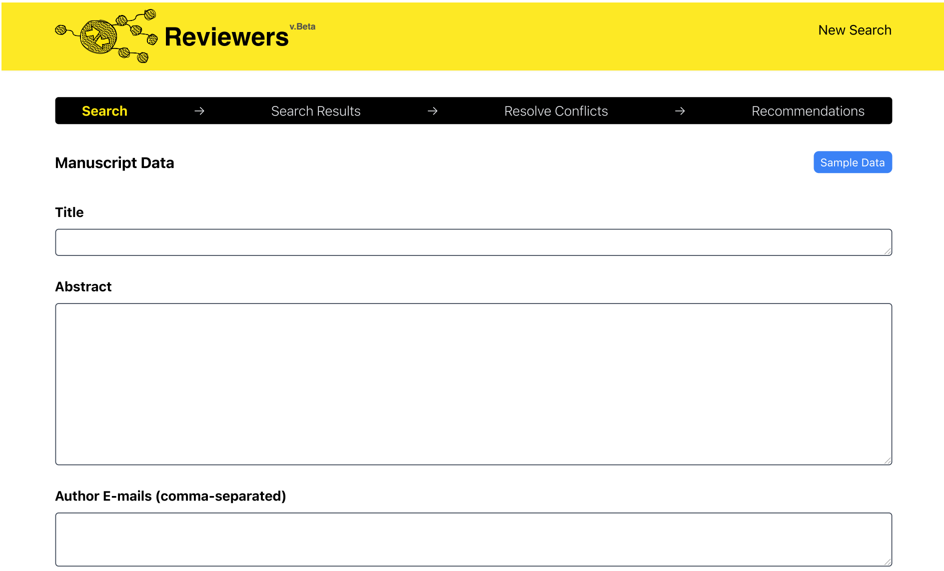

Figure 3: Prototype screenshot of the UI showing the data entry step.

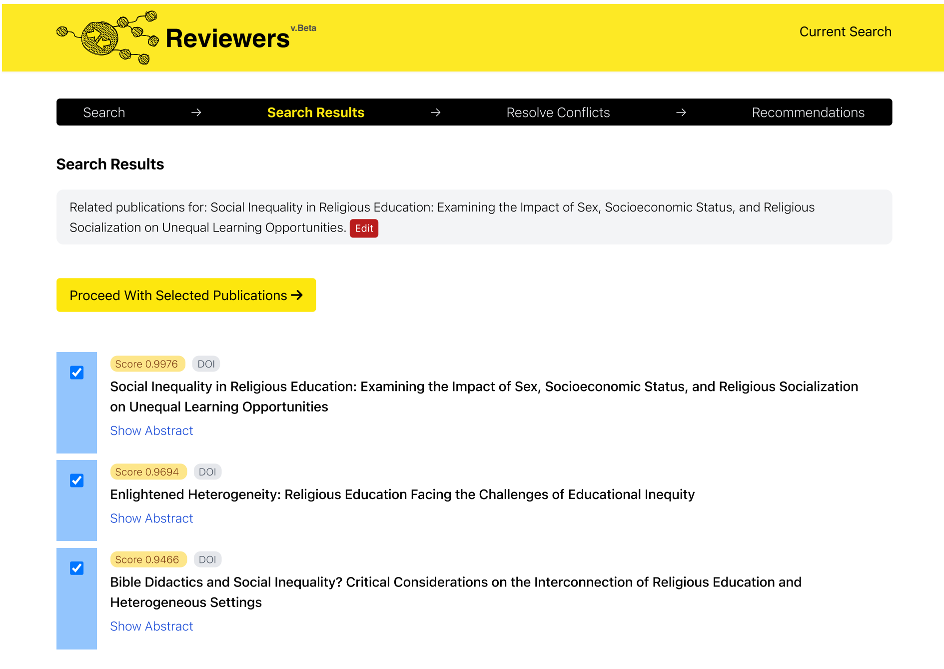

Figure 4: Prototype screenshot of the UI showing the related publication selection step.

3. In the third step, the list of proposed potential reviewers is presented along with the COIs that were identified. Additionally, background information such as the h-index, ORCID and current institute of the proposed reviewers are shown. Four flags are displayed

indicating a general status (has COIs / has no COIs) and the three types of COIs (Figure 5). Table 1 shows the author data and Table 2 shows the reviewer data and expected results for the COI resolution for the example manuscript shown in 5. The COIs displayed in the figure matched the expected ones. The first potential reviewer on top of the figure has no COIs, therefore the returned status is green. The identified COIs are represented with a red warning symbol. Consistently to the expected results, the COIs are shown for the remaining three potential reviewers.

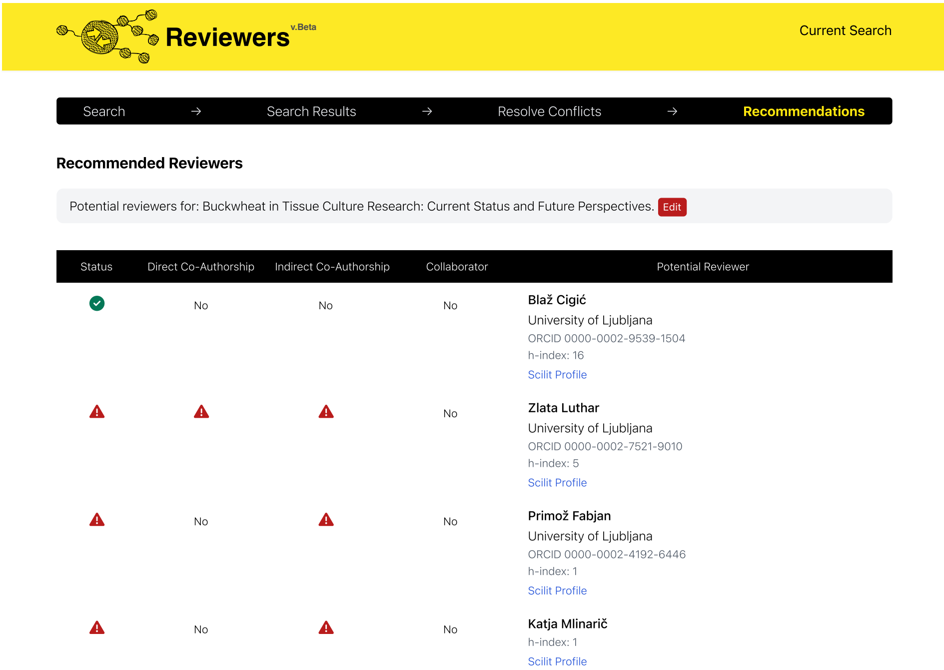

Figure 5: Prototype screenshot of the UI showing the proposed reviewers and the types of COIs that were identified.

Table 1
Author data for the example manuscript shown in Figure 5.

Author	Person Node	Institute Node
Alicja Tomasiak	Node 19314713	Node 853
Meiliang Zhou	Node 7658546	Node 54894
Alexander Betekhtin	Node 11401513	Node 853

6.1. Evaluation of the Perceived Usefulness and Usability of the Prototype

This section describes the conducted qualitative evaluation of the prototype. As criteria, we focused on the perceived usefulness and usability of the tool. The evaluation consisted of two phases. Firstly, the prototype was made available online to a panel of eight experts forming a

Table 2
Excepted results for the COI resolution from the example in Figure 5 and corresponding edges in the graph.

Reviewer Name	Person Node	Type & Source of COI
Zlata Luthar	Node 733659	Co-author of Zhou on 10.3390/plants10010014 Indirect co-author of Tomasiak and Betekhtin via Zhou
Primož Fabjan	Node 17935776	Indirect co-author of Zhou (Fabjan is co-author of Luthar, which is a co-author of Zhou via 10.3390/plants10081547)
Katja Mlinarič	Node 17935777	Indirect co-author of Zhou (Mlinarič is co-author of Luthar, which is a co-author of Zhou via 10.3390/plants10081547)

focus group of in-house MDPI editors[5]. The editors were asked to use the tool for three days as part of their daily work. Specifically, after they would assign reviewers to a manuscript with the as-is process, they should have used our tool for the same manuscript. Next, they had to note down a comparison between the two approaches. On the fourth day (i.e. second phase), we conducted a group and structured interview with all the editors for about 40 minutes. The perceived usefulness and usability of the tool were broken down into three sub-criteria: (1) the relevance of the matching papers to the topic of the manuscript, (2) the identification of COIs, (3) the user-friendliness and satisfaction in terms of speed of action-reaction of the user interface. The below sub-sections elaborate on each of the three sub-criteria.

6.1.1. The relevance of the matching papers to the topic of the manuscript

The interviewed editors agreed that the solution provided a quick way of matching related papers based on copying the title and the abstract of the manuscript. The results provided in the user interface through the STS matching were, despite the limited size of the database, highly relevant to the topics of the manuscripts tested. The proposed prototype was seen as advantageous compared to the keyword-based search. Editors could easily and automatically retrieve a list of relevant papers without the need for manual keyword identification. The editors particularly stressed the aspect of saving time. The number of suggested results (25 by default) was deemed sufficient. The possibility to uncheck some of the matched publications was welcomed as a way for editors to engage and interact with the prototype.

6.1.2. The identification of COIs

The effective identification of COIs is of high importance for the interviewed editors and is at the core of our proposed approach. Usually, editors working for MDPI journals are required to cover the literature of the past five years when searching for potential COIs. For our evaluation, the solution was loaded with the literature of two years only, 2020 and 2021. The reason is that it would have taken too long to transform and load literature data over five years (approx. 1 week) and there were no available server resources for that task. As a consequence, the

[5] https://reviewers.ch/, access credentials available upon request

identified COIs only focused on these two years. Nevertheless, there was a common agreement among the editors that the resulting COIs, although limited, were relevant and helpful for the identification of COIs when processing new manuscripts.

The interviewees also mentioned that it would be beneficial to load additional background data of the reviewers from other databases, such as Scilit or Google Scholar. As an example, one editor mentioned that adding the URL to the institutional homepage of the scholar would facilitate access to his/her background information to be checked by the editor.

Moreover, editors highlighted the need to take other criteria into account to assert the reliability and qualification of a potential reviewer. Current MDPI policies prescribe a minimum h-index, but interviewees mentioned that this may only be one of several possible filters. Knowing if the reviewer has already done reviews for MDPI in the past, and getting his/her score would be highly welcome. In the case of the h-index, a filter for a minimum value should allow for a flexible choice as in some research domains it could be higher and in others lower. A filter for a time span would also facilitate filtering out reviewers that have already been invited recently to review other manuscripts. As manual work is still necessary to check the background, the availability and the reliability of the referees, editors needed to rerun the search several times and suggested having a function to save the search results so that they can return to the search results and further refine it.

6.1.3. The user-friendliness and satisfaction in terms of speed of action-reaction of the user interface

Interviewed editors agreed on the simplicity and clarity of using the solution via the three-stepped approach in the user interface. All the editors found that the tool performed well and they could get the desired results quickly. One editor expressed the imminent desire to integrate the proposed tool into the existing manuscript processing system. Feedback for improvement was provided too. Interviewed editors agreed that the current prototype does not provide sufficient filters for narrowing-down the pool of qualified reviewer candidates. On one hand, editors would like the possibility to filter down the candidate pool to reviewers holding a PhD (or equivalent) degree. On the other hand, editors would like to narrow down the pool of reviewers by specifying a minimum h-index.

7. Discussion and Conclusions

In this paper, we presented a hybrid intelligent approach for the support of journal editors in the identification of Conflicts of Interest (COIs) of potential reviewers for a given manuscript. The approach was developed by following the Design Science Research methodology, where the design decisions were made based on the findings from both literature and expert interviews. The proof of concept demonstrated that the requirements could be met. From the evaluation with the focus group, it became clear that editors attach great importance to the requirement *R7*, i.e. the collaboration with the AI system. Ideally, a reviewer matching system should provide control to the editors and assist them in the quest to find domain experts that meet a number of qualification criteria. The system should automate certain tasks such as matching related literature, while giving more control to the editor to refine the search or narrow-down the pool of candidate reviewers. A direction for future improvements could be to allow editors to collect related publications into a publication pool first. They can then refine or expand using the same concept of STS matching of the items in the pool, before turning the publication pool into a pool of reviewer candidates. Future research directions could be of proving the approach in other academic journal publishers. An immediate next step could be of conducting additional evaluations by feeding the prototype with 5 years of literature and comparing the results with the ones from MDPI editors that use the as-is editorial process.

Acknowledgments

This work is a follow-up to a related project presented at the MAKEathon 2022[6]. We would like to thank the company Metaphacts[7] that originally provided the related challenge to the MAKEathon. Additionally, we would like to thank MDPI Scilit[8] for providing part of their disambiguate author data in order to build and evaluate the prototypical system. We would like to thank Dr. Daniele Masella for providing insights into the editorial process and the role of the meta-reviewers as part of an interview. Finally, we would also like to thank Yanping Mou, Carrie Guo, Chealsea Zhu, Fuli Cao, and Dr. Kero Dong for providing their feedback and evaluation as part of the focus group and an interview.

References

[1] O. Zimba, A. Gasparyan, Peer review guidance: a primer for researchers, Reumatologia/Rheumatology 59 (2021) 3–8. doi:10.5114/reum.2021.102709.

[2] F. Wang, B. Chen, Z. Miao, A Survey on Reviewer Assignment Problem, in: N. T. Nguyen, L. Borzemski, A. Grzech, M. Ali (Eds.), New Frontiers in Applied Artificial Intelligence, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2008, pp. 718–727. doi:10.1007/978-3-540-69052-8_75.

[3] K. Mittal, A. Jain, K. S. Vaisla, Understanding Reviewer Assignment Problem and its Issues and Challenges, in: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1–6. doi:10.1109/IoT-SIU.2019.8777727.

[4] C. J. Lee, C. R. Sugimoto, G. Zhang, B. Cronin, Bias in peer review, Journal of the American Society for Information Science and Technology 64 (2013) 2–17. doi:10.1002/asi.22784.

[5] D. B. Resnik, S. A. Elmore, Conflict of Interest in Journal Peer Review, Toxicologic Pathology 46 (2018) 112–114. doi:10.1177/0192623318754792.

[6] K. Li, Z. Cao, D. Qu, Fair Reviewer Assignment Considering Academic Social Network, in: L. Chen, C. S. Jensen, C. Shahabi, X. Yang, X. Lian (Eds.), Web and Big Data, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2017, pp. 362–376. doi:10.1007/978-3-319-63579-8_28.

[6] https://makeathonfhnw.ch/
[7] https://metaphacts.com/
[8] https://www.scilit.net/

[7] A. S. Nugroho, A. F. Iswafaza, R. N. E. Anggraini, R. Sarno, A Novel Approach on Conducting Reviewer Recommendations Based on Conflict of Interest, in: 2021 13th International Conference on Information & Communication Technology and System (ICTS), 2021, pp. 195–200. doi:10.1109/ICTS52701.2021.9609054.
[8] A. Cohan, S. Feldman, I. Beltagy, D. Downey, D. S. Weld, Specter: Document-level representation learning using citation-informed transformers, 2020. doi:10.48550/ARXIV.2004.07180.
[9] D. T. Hoang, N. T. Nguyen, D. Hwang, A Group Recommender System for Selecting Experts to Review a Specific Problem, in: 10th International Conference, ICCCI 2018, Bristol, UK, September 5-7, 2018, Proceedings, Part I, 2018, pp. 270–280. doi:10.1007/978-3-319-98443-8_25.
[10] X. Zhao, Y. Zhang, Reviewer assignment algorithms for peer review automation: A survey, Information Processing & Management 59 (2022) 103028. doi:10.1016/j.ipm.2022.103028.
[11] N. Kotak, A. K. Roy, S. Dasgupta, T. Ghosal, A Consistency Analysis of Different NLP Approaches for Reviewer-Manuscript Matchmaking, in: H.-R. Ke, C. S. Lee, K. Sugiyama (Eds.), Towards Open and Trustworthy Digital Societies, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 277–287. doi:10.1007/978-3-030-91669-5_22.
[12] D. Yarowsky, R. Florian, Taking the load off the conference chairs-towards a digital paper-routing assistant, in: 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, pp. 220–230. URL: https://aclanthology.org/W99-0627.
[13] S. Hettich, M. J. Pazzani, Mining for proposal reviewers: lessons learned at the national science foundation, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, Philadelphia PA USA, 2006, pp. 862–871. URL: https://dl.acm.org/doi/10.1145/1150402.1150521. doi:10.1145/1150402.1150521.
[14] L. Charlin, R. Zemel, The Toronto Paper Matching System: An automated paper-reviewer assignment system, in: ICML 2013 Workshop on Peer Reviewing and Publishing Models, 20 June 2013, Atlanta, Georgia, USA, 2013, p. pp. 9. URL: https://openreview.net/forum?id=caynafZAnBafx.
[15] D. Mimno, A. McCallum, Expertise modeling for matching papers with reviewers, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 500–509. doi:10.1145/1281192.1281247.
[16] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, 2019. doi:10.48550/ARXIV.1903.10676.
[17] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, IEEE Transactions on Big Data 7 (2019) 535–547.
[18] M. Aumüller, E. Bernhardsson, A. Faithfull, Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms, Information Systems 87 (2020) 101374. URL: https://www.sciencedirect.com/science/article/pii/S0306437918303685. doi:https://doi.org/10.1016/j.is.2019.02.006.

[19] C. Long, R. C.-W. Wong, Y. Peng, L. Ye, On good and fair paper-reviewer assignment, in: 2013 IEEE 13th International Conference on Data Mining, 2013, pp. 1145–1150. doi:10.1109/ICDM.2013.13.

[20] N. B. Shah, Challenges, experiments, and computational solutions in peer review, Communications of the ACM 65 (2022) 76–87. doi:10.1145/3528086.

[21] M. Färber, L. Ao, The Microsoft Academic Knowledge Graph enhanced: Author name disambiguation, publication classification, and embeddings, Quantitative Science Studies 3 (2022) 51–98. URL: https://direct.mit.edu/qss/article/3/1/51/109628/The-Microsoft-Academic-Knowledge-Graph-enhanced. doi:10.1162/qss_a_00183.

[22] Y. Deng, Recommender Systems Based on Graph Embedding Techniques: A Review, IEEE Access 10 (2022) 51587–51633. doi:10.1109/ACCESS.2022.3174197.

[23] S. Verma, R. Bhatia, S. Harit, S. Batish, Scholarly knowledge graphs through structuring scholarly communication: a review, Complex & Intelligent Systems (2022). doi:10.1007/s40747-022-00806-6.

[24] P. Ristoski, J. Rosati, T. Di Noia, R. De Leone, H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, Semantic Web 10 (2019) 721–752. URL: https://content.iospress.com/articles/semantic-web/sw317. doi:10.3233/SW-180317.

[25] A. Hevner, S. Chatterjee, Design Research in Information Systems, volume 22 of *Integrated Series in Information Systems*, Springer US, Boston, MA, 2010. doi:10.1007/978-1-4419-5653-8.

[26] L. Ham, Introduction to weaviate vector search engine, 2021. doi:10.5281/zenodo.4903211.