IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2021, accepted January 23, 2022, date of publication February 15, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151640

Reviewer Recommendations Using Document Vector Embeddings and a Publisher Database: Implementation and Evaluation

YUE ZHAO[1], AJAY ANAND[2], AND GAURAV SHARMA[3], (Fellow, IEEE)
[1]Rochester Data Science Consortium, Rochester, NY 14607, USA
[2]Goergen Institute for Data Science, University of Rochester, Rochester, NY 14627, USA
[3]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA

Corresponding author: Gaurav Sharma (g.sharma@ieee.org)

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1934962.

ABSTRACT We develop and evaluate an automated data-driven framework for providing reviewer recommendations for submitted manuscripts. Given inputs comprising a set of manuscripts for review and a listing of a pool of prospective reviewers, our system uses a publisher database to extract papers authored by the reviewers from which a *Paragraph Vector* (*doc2vec*) neural network model is learned and used to obtain vector space embeddings of documents. Similarities between embeddings of an individual reviewer's papers and a manuscript are then used to compute manuscript-reviewer match scores and to generate a ranked list of recommended reviewers for each manuscript. Our mainline proposed system uses full text versions of the reviewers' papers, which we demonstrate performs significantly better than models developed based on abstracts alone, which has been the predominant paradigm in prior work. Direct retrieval of reviewer's manuscripts from a publisher database reduces reviewer burden, ensures up-to-date data, and eliminates the potential for misuse through data manipulation. We also propose a useful evaluation methodology that addresses hyperparameter selection and enables indirect comparisons with alternative approaches and on prior datasets. Finally, the work also contributes a large scale retrospective reviewer matching dataset and evaluation that we hope will be useful for further research in this field. Our system is quite effective; for the mainline approach, expert judges rated 38% of the recommendations as *Very Relevant*; 33% as *Relevant*; 24% as *Slightly Relevant*; and only 5% as *Irrelevant*.

INDEX TERMS Reviewer matching, text mining, document vector embedding, evaluation methodology, explainable learning.

I. INTRODUCTION

To assess manuscripts submitted for publication, scientific journals and conferences invariably rely on peer-review, i.e., on assessments of the work provided by other researchers having expertise on the topic of the manuscript. Traditionally, the process of identifying expert reviewers has been manual, relying on the journal editorial board members' or conference technical program committee members' familiarity with the research community in their areas. The manual approaches are increasingly encountering challenges of scale due to increases in the size and geographic-spread of the research communities and the accompanying increase in manuscript submission volumes. As a result, there is growing interest in automating reviewer assignment using modern computational and machine learning methodologies.

In its entirety, reviewer assignment is a complex task that must take into account multiple objectives and constraints. Invariably, the process begins with the identification of prospective reviewers with expertise that is matched with a submitted manuscript. Additionally, reviewer assignment must also ideally ensure coverage of different technical areas that contribute to a manuscript's novelty/innovation, compliance with individual reviewers' workload restrictions, exclusions due to conflicts of interest, and fairness and thoroughness of assessment. Instead of attempting an exhaustive survey, we refer the reader to [1] for an excellent overview that highlights the different tasks involved in the complete

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

reviewer allocation process and the various approaches/tools available. In this paper, we focus on the key first step in the reviewer assignment process. Specifically, we examine a data-driven approach for identifying a ranked list of prospective reviewers for manuscripts and, with a view to providing better guidance for this subtask, evaluate alternative choices and options for this purpose.

Manuscript-to-reviewer matching is traditionally based on a categorical matching methodology. A set of topics/keywords is used to partition both manuscripts and reviewer expertise into a common set of classification categories and the suitability of a manuscript-reviewer match is assessed based on the category classes they share. The category driven approach has severe shortcomings: the topics/keywords require frequent updates, invariably fail to provide complete coverage, while also suffering from significant overlaps that limit their effectiveness for scope delineation. Authors and reviewers feel strait-jacketed in having to assign their manuscript to specific categories, particularly for work that is multidisciplinary or exploring entirely new directions that do not readily fit in the existing categorization scheme. Categories also can be quite heterogeneous in size, resulting in too many or too few matches for a manuscript/reviewer and manuscript authors, and reviewers can differ widely in the number of topics/keywords that they choose. Data-driven approaches that do not rely on pre-assigned categories are therefore of interest and these are the focus of our study.

With scholarly data becoming available at scale (see [2] for a broader discussion and context on scholarly big data), a number of other researchers have utilized data-driven approaches for reviewer matching. At their core, these techniques rely on assessing the similarity between a manuscript submitted for review and prior papers authored by prospective reviewers, based on which, a reviewer match score is computed. A number of text mining techniques have been used for assessing the similarity between the manuscript and paper documents for this purpose. The relatively simple bag-of-words model, where the similarity between documents can be assessed based on the common words they share has been used [1], [3] for reviewer matching with the term frequency-inverse document frequency (TF-IDF) weighting [4]. Reviewer matching methods have also been developed based on document similarity assessments from probabilistic text modeling techniques. Instead of using pre-assigned topics, these techniques estimate both a set of topics from a document corpus and the distribution of each document over the topics, which is then used to compute document similarities. Specifically, reviewer matching has been performed using document similarity assessments from Latent Dirichlet Allocation (LDA) [5], [6], Latent Semantic Indexing (LSI) [7], and Probabilistic Latent Semantic Indexing (PLSA) [8]. In a variant of this methodology, an Author-Persona-Topic [9] has also been proposed that allows for alternative personas for reviewers to account for multifaceted expertise/research areas. Apart from one notable exception, the majority of these prior techniques use only the abstracts of the reviewers' published papers in the matching process, as the abstracts are obtainable publicly from bibliographic databases and/or indexing services without requiring a subscription to the published content. The Toronto Paper Matching System (TPMS) [6], which is the exception, uses full text papers in portable document format (PDF) provided by the reviewers. Getting reviewers to participate and to upload a representative set of their papers makes this approach more challenging to manage and sustain. Additionally, there is also potential for malicious manipulation through misrepresentation by reviewers and fake reviewing accounts. These ethical concerns are not purely speculative fear-mongering as incidents of fake reviewer rings have plagued multiple publishers of scientific literature [10]–[12].

In this paper, we attempt to address several of the above mentioned challenges in prior systems. First, we implement a prototype data-driven system for manuscript-reviewer matching leveraging more recent data driven models, specifically, the *doc2vec* document vector space embeddings [13], which have been shown to be very effective in determining semantic similarity of texts from different domains and having different lengths [14]–[16]. In this context, our particular interest is in assessing and providing guidance on how the richness of the underlying data impacts the quality of the matching. Specifically, we compare models developed based only on abstracts, which represent the dominant paradigm in past work, against those developed based on the full text from the papers, where the larger data-size enables richer and more effective representations. Second, our system extracts papers authored by the reviewers directly from a publisher database without requiring input from the reviewers. This approach not only alleviates the burden on the reviewers and the editorial board/program committee but also ensures that the underlying data is up-to-date and reliable, and eliminates the potential for unethical manipulation of the peer review process via data manipulation. Third, we present a thorough evaluation of our system using a retrospective dataset, and, in the process, also contribute an evaluation methodology and a dataset useful for future work in this area. Our proposed multi-stage evaluation methodology provides: (a) a self-consistency test that informs model hyperparameters, (b) a primary evaluation that builds upon the approach in [9] and eases the burden on expert judges assessing the matches by providing them an effective user interface, and (c) a secondary evaluation approach that allows us to indirectly compare against alternative approaches and choices, and also assess performance on other datasets. Through the combination of these three elements, our evaluation methodology specifically addresses the challenges of assessing reviewer matching recommendations at scale in the absence of a unique answer and without exhaustively labeled datasets, which are infeasible to obtain in realistic settings.

The rest of the paper is organized as follows. Section II describes our reviewer matching system and provides details of the components used in its realization. In Section III, we propose our evaluation methodology. Section IV presents our experiments and results, including details of the primary

dataset we created for our mainline methodology, comparisons against alternative approaches and choices, and indirect evaluation on an available prior dataset. Section V discusses the limitations of our study. Finally, Section VI concludes the paper with a summary of the main findings. An appendix summarizes relevant background information on the *doc2vec* model that is at the core of our methodology for assessing reviewer to manuscript matches.

II. REVIEWER MATCHING SYSTEM

Figure 1 provides an overview of our proposed approach, which uses, as inputs, a list of manuscripts (or their abstracts) intended for peer-review and a list of prospective reviewers, and outputs a ranked list of recommended reviewers for each manuscript. The approach has two main stages, both of which access relevant data from a publisher database. In the first stage, an archetypal training corpus of papers is obtained from the publisher database, which is then used to train a *doc2vec* model for mapping documents into fixed length vectors representative of their content. The resulting trained model is defined by the model parameters determined in the training process, which are then used in the second (matching) stage. In the matching stage, first, using the reviewer corpus, the publisher database is queried to obtain the papers authored by each reviewer and using the trained *doc2vec* model from the first stage, each of these papers is mapped to a corresponding vector. The manuscript under consideration is similarly mapped via the trained *doc2vec* model into a corresponding manuscript vector. Then, using the vector representations, a pairwise similarity is computed between the manuscript vector and each of the reviewer paper vectors. These pairwise similarity scores are then consolidated to determine an appropriateness score for each reviewer for the manuscript. The rank ordered list of reviewers based on the score is then computed and is the output of the matching process. In the remainder of this section, we detail the two main stages in the proposed approach, explaining our motivations for the design choices and highlighting remaining parameters whose values are subsequently determined empirically.

A. DOC2VEC MODEL CHOICE AND MODEL TRAINING

Fixed length vector representations of documents have become an essential component of document retrieval and clustering tasks. Documents are mapped to fixed length document vectors, which then allow for a variety of operations to be conducted using the vector space notions of distance and similarity. Among such approaches, *doc2vec* [13] represents a powerful current technique, where the mapping from documents to corresponding vectors is accomplished using a fully-connected two-layer feed-forward neural network trained over a representative corpus of documents. For a given document, using the trained network, one can compute a fixed length vector that predicts the probabilities of words that are in the document and words from the corpus that are not in the document. To make this article self-contained, we provide a high level overview of *doc2vec* and its closely associated predecessor *word2vec* [17] in the appendix, where we particularly highlight architectural choices and options used in our work.

Document vectors obtained with *doc2vec*, and word vectors obtained with its closely associated predecessor *word2vec* [17], have been shown to produce state-of-the-art results in a number of text mining tasks. Document vectors from *doc2vec* have been successfully used in article clustering [13], [14], semantic similarity estimation between sentence pairs [15], [16], and query de-duplication [18].

At its core, effective reviewer matching relies on document clustering; most suitable reviewers are individuals who have authored papers that would lie in the same clusters as the manuscript under consideration. Thus, similarity scores from the *doc2vec* model should be well suited to identifying recommended reviewers. Figure 1 illustrates the training process.[1] A training corpus of papers is obtained by retrieving papers from the publisher database. These papers should be representative of the technical areas for the journal/conference for which reviewer recommendations are being sought and the problem setting often determines a suitable corpus. For example, for a conference, the pool of papers authored by the reviewers is a natural training corpus. For journals, the training corpus can comprise not only the papers authored by the journal's reviewer pool and editorial board but can also include papers published over the past several years in the journal and those in closely affiliated conferences. After preprocessing to remove non-alphabetical characters, conversion to lower-case, and stemming, text from the training corpus can then be used to train the *doc2vec* model for subsequent use. Alternative choices can be made in this process to either use the full text from the papers for developing the model or to use only the abstract text. We call a model trained on the full text corpus the full text model (*full-model*) and a model trained on the abstracts the abstract model (*abs-model*). Full text offers the potential for more comprehensive and richer representations whereas training based on abstracts alone can reduce training time (and model size) and has key advantage that abstracts are much more broadly available without requiring a subscription. In this process, the dimension N of the vector representations provided by the model is a hyperparameter choice that needs to be made based on the size of the training corpus and its impact is empirically explored in subsequent sections. The training process yields parameters θ for the trained model that constitute the weights for the feedforward neural network and which are later used in the matching process.

B. REVIEWER MATCHING

The second stage, reviewer matching, has three components as shown in Fig. 2. First, we retrieve the text of the prospective reviewers' papers using a publisher database. The list of the prospective reviewers is denoted by

[1] In this and other figures in this paper, rounded rectangles represent entities and rectangles with sharp corners represent processing steps.

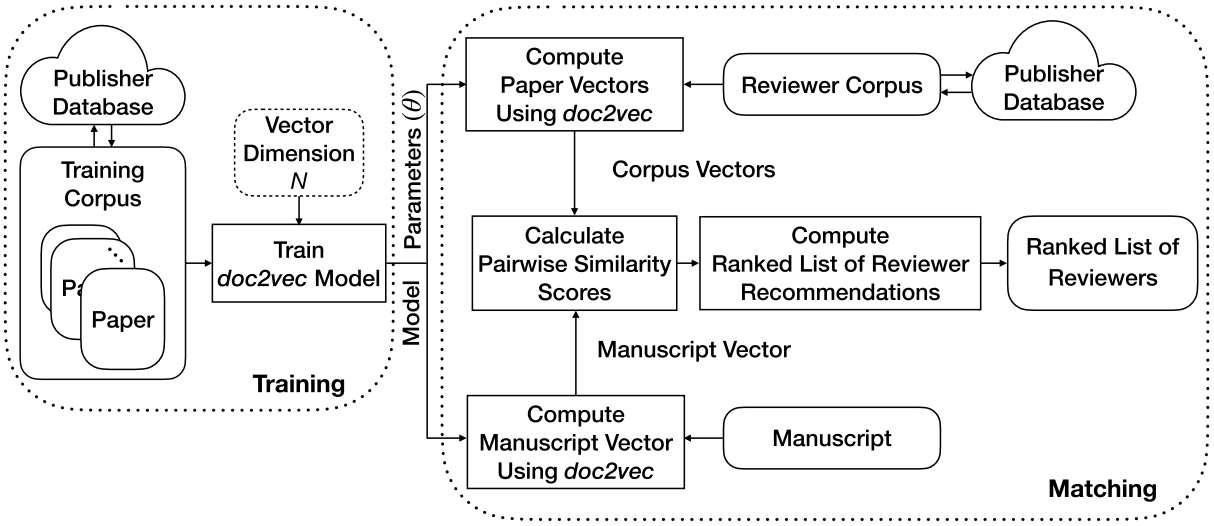


FIGURE 1. Overview of the proposed reviewer matching approach.

$\mathcal{R} = \{R_1, R_2, \ldots, R_L\}$. We make an effort to disambiguate reviewers' name by entering their affiliation, in addition to their full name. A reviewer's unique ID number is used whenever possible because it offers better and easier disambiguation. The preprocessed reviewers' papers are denoted by $\mathcal{C} = \{C_1^1, C_2^1, \ldots, C_{m_i}^i, \ldots, C_{m_L}^L\}$, where the superscript i denotes reviewer R_i, and the subscript m_i is the number of papers for reviewer R_i that are retrieved from the publisher database. Then the trained *doc2vec* model is used to compute the paper vectors, $\mathcal{D} = \{\mathbf{d}_1^1, \mathbf{d}_2^1, \ldots, \mathbf{d}_{m_i}^i \ldots, \mathbf{d}_{m_L}^L\}$, where $\mathbf{d}_j^i$ is the vector representation of the j^{th} paper of reviewer R_i.

The preprocessed manuscripts, denoted by $\mathcal{M} = \{M_1, M_2, \ldots, M_k, \ldots\}$, are also mapped into corresponding manuscript vectors $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k, \ldots\}$ using the trained *doc2vec* model, where the dimensions of the vectors are identical to those for the paper vectors. When computing the manuscript vectors, words that have never appeared in model training are ignored [19]. For each manuscript M_k and each document vector C_j^i in $\mathcal{C}$, a normalized similarity score in the range $[-1, 1]$ is then computed as the cosine similarity

$$\text{sim}(M_k, C_j^i) = \frac{\mathbf{v}_k^\top \mathbf{d}_j^i}{|\mathbf{v}_k||\mathbf{d}_j^i|}, \tag{1}$$

between the corresponding document vectors $\mathbf{v}_k$ and $\mathbf{d}_j^i$, where a larger value indicates greater similarity. Similarity scores above a threshold κ are then used to compute a match score for each reviewer with the manuscript. Specifically, the match score $S(M_k, R_i)$ for reviewer R_i with the manuscript M_k is computed as

$$S(M_k, R_i) = \begin{cases} \left(\sum_{j \in \mathcal{T}_i^k} \text{sim}^p \left(M_k, C_j^i \right) \right)^{\frac{1}{p}} & \mathcal{T}_i^k \neq \phi \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

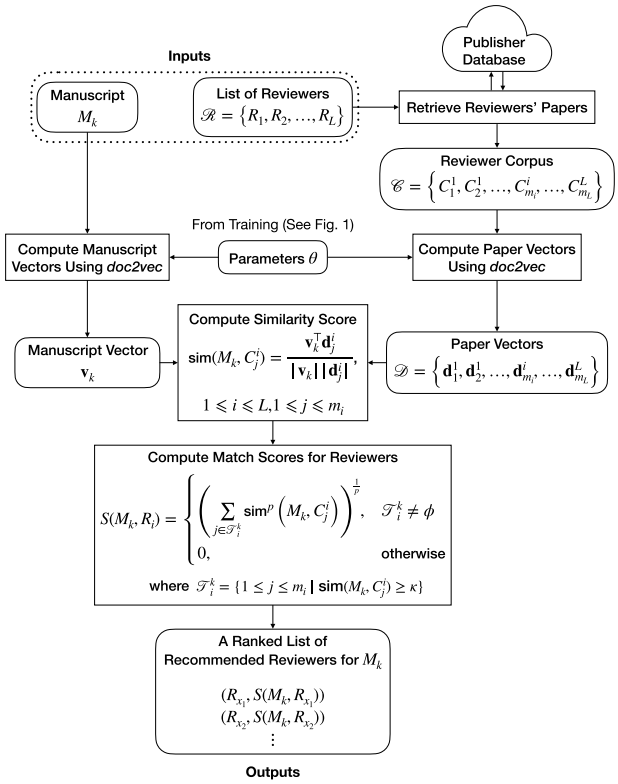


FIGURE 2. Reviewer matching process using a pretrained *doc2vec* model.

where

$$\mathcal{T}_i^k = \{1 \leq j \leq m_i \mid \text{sim}(M_k, C_j^i) \geq \kappa\} \tag{3}$$

is the set of manuscript indices for reviewer R_i whose similarity score with the manuscript M_k exceeds the threshold κ, ϕ denotes the null set, and p is a positive real parameter. The similarity threshold κ eliminates reviewer papers that are only marginally similar to the manuscript M_k from the

computation of the reviewer match score thereby preventing a reviewer who has a large number of only marginally relevant papers from accumulating a large match score. The expression in the first line on the right hand side of (2) corresponds to the p-norm of the similarity scores above the threshold κ. As with the p-norm, the choice of the parameter p determines the relative emphasis between the smaller and larger valued similarity scores $\text{sim}(M_k, C_j^i)$ included in the summation. A larger value of p emphasizes the contribution of the larger valued scores and as $p \to \infty$, $S(M_k, R_i)$ approaches $\max_{j \in \mathcal{T}_i^k} \text{sim}(M_k, C_j^i)$, so that only the one paper for the reviewer with the best match contributes to the reviewer match score. On the other hand, a smaller value of p reduces the impact of the similarity score's numerical values on the final reviewer match score and as $p \to 0$, $S(M_k, R_i)$ approaches the number $|\mathcal{T}_i^k|$ of elements in $\mathcal{T}_i^k$, so that all similarity values (above κ) count equally in the reviewer match score computation.

The combination of the threshold κ and the parameter p also determine the influence (on the reviewer's match score) of the number of publications that prospective reviewers have related to the manuscript. Exclusion of papers with similarity scores below κ prevents a high reviewer score from being accumulated based simply on a large number of marginally relevant manuscripts and, as already noted, a value of p much larger than 1 emphasizes the contribution of reviewer papers that have a higher score in the computation of the overall reviewer score.

For each manuscript M_k, a ranked list of recommended reviewers is then output by ordering the reviewers in descending order by their match scores for the manuscript. In this process, any reviewers whose names match those of any of the authors on the manuscript are removed so that we do not recommend an author of the manuscript to be a reviewer of that manuscript.

III. EVALUATION METHODOLOGY

Evaluation of reviewer recommendations is challenging for several reasons. Reviewer matches with a manuscript are inherently subjective and non-unique. Several alternative reviewers may be well-matched with a given manuscript. Typically, one relies on expert judges for evaluating manuscript-reviewer matches. The judges need specialized technical knowledge covering both breadth and depth of topics represented in the manuscripts and the papers; crowd-sourcing evaluations is therefore not a viable option. Assessment of appropriateness or ranking of each manuscript-reviewer pair by a qualified judge would allow for the creation of comprehensive *a priori* "ground truth" against which algorithmically generated assessments could be evaluated. However, extensive time requirement from judges and the tedium involved in such an approach render it untenable for datasets that have adequate size and diversity to be meaningful. Therefore, in practice, the evaluation of a reviewer recommendation framework is typically done by having a panel of expert judges manually rate the framework's recommendations for a subset of randomly selected manuscripts. This approach still demands considerable time from the judges, so the methodology cannot be directly used to also perform assessments of alternative parameter choices and options for manuscript-reviewer matching algorithms.

To tackle the afore-mentioned challenges we propose a hybrid multi-stage evaluation framework that effectively integrates automatic and manual assessments. Additionally, we also developed a simple user interface to allow judges to preform their assessments and provide ratings without too much tedium. The evaluation methodology and user interface also constitute key contributions of the present work. The first stage of our framework is an automated consistency test that allows us to explore alternative design choices and to select appropriate values for the algorithmic parameters. Next we present our primary evaluation methodology based on manual assessments, from a panel of expert judges, of the recommendations provided by our framework for an example dataset. Then we present additional methods that leverage the relevance ratings provided by expert judges in a secondary evaluation that allows us to partly compare our approach with a more traditional LDA model based approach for manuscript-reviewer matching. Finally, we present an additional validation approach that allows us to indirectly evaluate our approach on a publicly available dataset. We detail each of these stages in the following subsections.

A. CONSISTENCY TEST

Document vectors for a trained *doc2vec* model are obtained via an optimization procedure using the fully-connected two-layer neural network that comprises the model. Thus recomputations of the document vectors for a manuscript can yield different document vectors.

When the vector dimension N is chosen to be appropriately large, we expect alternative computations of the document vectors for the same document to be similar and invariably closer together than the document vectors obtained from different documents. Furthermore, in the specific context of technical papers, one also expects that, for a robust model, the document vectors for the full text document and the corresponding abstract should be closer to each other than those for different documents. Let $\mathbf{d}_j^i$ and $\mathbf{d}_j'^i$ represent alternative document vectors for the document C_j^i for $i \in \{1, 2, \ldots, L\}$, $j \in \{1, 2, \ldots, m_i\}$ where one of these could be obtained from the training process itself. Then, by the preceding argument, we expect the *self-similarity* $\text{sim}(\mathbf{d}_j'^i, \mathbf{d}_j^i)$ for the pair of alternative document vectors $\mathbf{d}_j^i$ and $\mathbf{d}_j'^i$ for the same document to be overwhelmingly the top ranked similarity among the pairwise similarities $\text{sim}(\mathbf{d}_{j'}'^{i'}, \mathbf{d}_j^i)$ arranged in decreasing order for all $i, i' \in \{1, 2, \ldots, L\}$, and $j, j' \in \{1, 2, \ldots, m_i\}$. Furthermore, for either the *full-model* trained on full-text data or the *abs-model* trained on abstracts-only, we expect the rank similarity to hold for document vectors computed either from the full-text or the abstract. The consistency test is completely automated and does not require human assessment and the

percentage of self-similar ranks that is larger than K, i.e. not in the top matches for a small value of K, provides an indication for what is appropriate for the document vector dimension N, allowing this to then be set for the subsequent experiments.

B. PRIMARY EVALUATION

For the primary evaluation of our reviewer recommendations, we select a subset of manuscripts randomly and invited expert judges to provide their assessment of the relevance of the recommendations from our proposed approach. The judges provided their assessments via a Python-based graphical user interface (GUI) that we created for this purpose. The GUI significantly simplified the judges' task by allowing them to work sequentially through the manuscript-reviewer matches for a manuscript, one by one for each of the manuscripts in the evaluation subset. Explainability of the matching is key to human assessment of the appropriateness. Therefore, each manuscript was presented along with its corresponding ranked list of recommended reviewers and upon clicking on a reviewer, relevant information for making the assessment was made available on the GUI screen in a split view mode as illustrated by the example screenshot in Fig. 3. The left half presented the manuscript information with the title and abstract shown and a clickable link to access the manuscript PDF document. The right half showed relevant reviewer information with the reviewer name and affiliation and ranked list of their papers based on the similarity match with the manuscript along with the abstracts and a clickable link to access the paper's record on the publisher's website. The judges relevance ratings entered via the interface were recorded in an XML file by the GUI, which was then provided by the judges to us once the assessments were completed.

For each recommended reviewer, the judge was asked to enter in their assessment of the match relevance on a four-level 0–3 integer scale established by Mimno & McCallum [9] and we also adopted their instructions to the judges for the assessments, which instructed the judges to assess relevance as follows.

The four levels of relevance, adopted from Mimno & McCallum [9] are defined as follows:

- *Very Relevant* (3): All areas of the manuscript are covered by the reviewer's papers that are listed (in the matching results). For example, suppose the manuscript is about areas {A, B, C}. If the best matched papers of the reviewer collectively cover {A, B, C}, then the reviewer is considered *Very Relevant* for the manuscript.
- *Relevant* (2): Most areas of the manuscript are covered by the reviewer's papers that are listed. For example, suppose the manuscript is about areas {A, B, C}. If the best matched papers of the reviewer collectively cover {A, B} or any other two areas, then the reviewer is considered *Relevant* for the manuscript.
- *Slightly Relevant* (1): Few areas of the manuscript are covered by the reviewer's papers that are listed. For example, suppose the manuscript is about areas {A, B, C}. If the best matched papers of the reviewer collectively cover A or one of the other areas, then the reviewer is considered *Slightly Relevant* for the manuscript.
- *Irrelevant* (0): No areas of the manuscript are covered by the listed reviewer's papers.

The primary evaluation was performed based on the *full-model* obtained from training the *doc2vec* model on the full-text data for the reviewer paper corpus.

C. COMPARISON AND SECONDARY EVALUATION

Despite our GUI for streamlining the evaluation process, the primary evaluation requires significant time commitment from the judges, and it is not feasible to get direct additional evaluations for alternative choices and approaches that we wish to compare against. Therefore, to provide such comparison, albeit limited, we rely on secondary analysis which reuses the judges' ratings from the primary evaluation. Specifically, we evaluated alternative approaches by treating the primary evaluation relevance values as "ground-truth" and computing the recall at K for the alternative approaches as

$$\text{Recall}_\rho(K) = \frac{P_\rho^K}{Q_\rho}, \tag{4}$$

where for a given relevance value ρ, P_ρ^K is the number of recommendations rated at relevance ρ in the ground truth that are ranked in the top K options by the alternative approach, and Q_ρ is the total number of recommendations rated at relevance ρ in the ground truth. If the alternative technique performs well, ideally, the matching rank of the alternative technique should have a strong (negative) correlation with the relevance rating, i.e., more relevant matches should be ranked higher and included among the top recommendations. Conversely, reviewers who are rated as *Irrelevant* should not appear in the top recommendations, though we need to keep in mind the caveat that the judges' evaluations used as "ground truth" are limited to the matches presented to them for the proposed approach. The value of K determines the tolerance in the matching and meaningful values will depend on the application constraints and downstream usage. For instance, in situations where an expert human intermediary is using the ranked lists, a relatively small value of K could provide the expert options to investigate further and choose between.

We note that the methodology of using Recall at K can also be used to assess the proposed technique against relevance evaluation datasets obtained for other reviewer recommendation techniques, specifically, for the dataset and relevance ratings gathered by Mimno and McCallum [9] and we also perform this secondary evaluation for the recommendations provided by our framework based on the *abs-model* (only abstracts are available in this dataset).

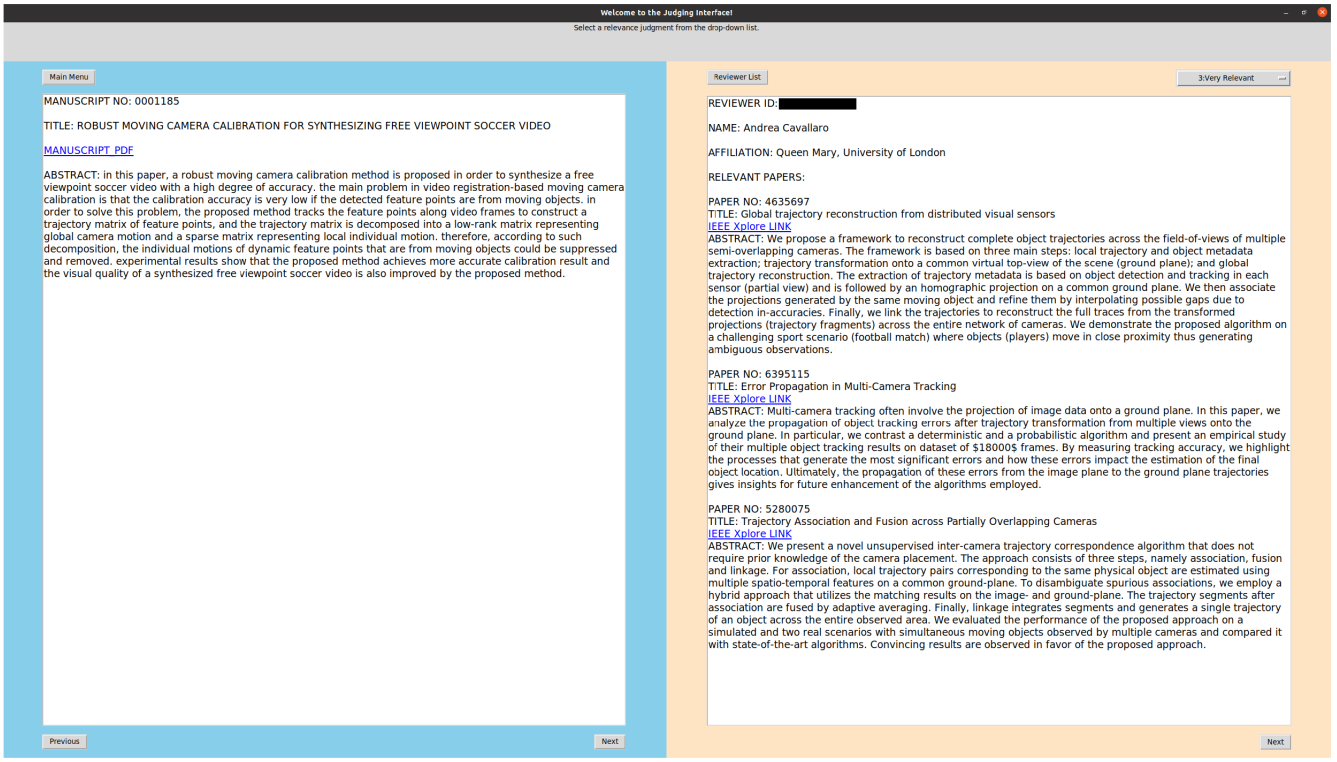


FIGURE 3. Main GUI screen for judge's assessment of manuscript-reviewer match relevance and examination of relevant data. The left pane shows the manuscript information and the right pane shows the information for an identified matching reviewer and the reviewers' papers contributing to the match score, in decreasing order of contribution.

IV. EXPERIMENTS AND RESULTS

Our proposed approach was implemented on a full-scale dataset which we detail first in Subsection IV-A. Next, in Subsection IV-B we describe the choice of model hyperparameters, which is informed, in part, by the consistency test based evaluation. We then present the results from the primary evaluation in Subsection IV-C and the secondary evaluation based comparison with other techniques in Subsection IV-D.

A. RETROSPECTIVE IEEE ICIP 2016 DATASET

For a full-scale implementation of our proposed framework, we used publicly available data[2] from the 2016 edition of the *IEEE International Conference on Image Processing* (*ICIP*), which is a long-standing flagship conference of the IEEE Signal Processing Society and for which one of us served as a Technical Program Co-Chair. The published reviewer names and affiliations for the conference served as our reviewer list $\mathcal{R}$ and we obtained the corpus of reviewers' papers $\mathcal{C}$ from the IEEE Xplore digital library [20]. Specifically, using the IEEE Xplore API and SDK [21] we obtained records for papers published prior to the year 2016 for which at least one of the author names matched one of our reviewer names. These paper records were then filtered based on matching the authors' affiliation to eliminate spurious matches. Also, reviewers who had fewer than five papers available on Xplore database were excluded due to indication of inexperience and lack of representative data. In situations where the reviewer database included multiple authors for a paper, only one unique instance of the paper was retained.[3] Abstracts for the reviewer corpus $\mathcal{C}$ were also obtained in plain text format using the API provided by the SDK, and corresponding full-text PDF documents were provided to us by the IEEE Xplore team upon request. Text was extracted from the PDF documents using pdftotext [22] and results with fewer than 1,200 English words were dropped to filter out erroneous PDF-to-text conversions and documents that were not technical papers. The final dataset consisted of 67,461 full texts and 66,994 abstracts from 1,833 reviewers.[4] After preprocessing to remove non-alphabetical characters, conversion to lowercase, stemming, and elimination of words that occur very infrequently in the entire corpus (less than 15 times for the full-texts which were prone to noisy characters from the PDF conversion and less than 2 times for the abstracts), the vocabulary size was 96,743 words for the full texts and 15,723 words for the abstracts.

[2]Access to the full-text papers requires a subscription to IEEE Xplore.

[3]In such cases, the paper contributed only once to the training of the *doc2vec* model but was included in the computation of match scores for each of the reviewer co-authors as per the methodology outlined in Section II-B

[4]The API we used to extract abstracts for the papers from Xplore found some of the abstracts missing or incomplete whereas the Xplore team provided us an almost complete set of full-text PDFs. As a result, we ended up with a few more full texts than abstracts.

B. MODEL HYPERPARAMETERS AND CONSISTENCY TEST

For implementing our proposed framework, we used the Gensim [23] implementation of the *doc2vec* algorithm. We chose to use the distributed bag-of-words (DBOW) model, which is shown to have better performance [18]. While using DBOW, we turned on the option of simultaneously training the word vectors in the DBOW model[5] since it has been reported that the document vectors achieve better quality when word vectors are updated jointly [14], [18] during training. For this purpose, the window size hyperparameter was set to 10. The number of negative words sampled is set to 5 and the down-sampling threshold is 10^{-5} as suggested by [18]; the learning rate was set constant at 0.05 and training used 30 epochs.

We experimented with different values of the vector dimension N for the *full-model* and the *abs-model* and used the consistency test evaluation described in Section III-A to assess relative performance. Figure 4 shows the results of the model consistency test described in Section III-A, where, for $K = 1$ and $K = 4$, the plots show the percentage of document self-similarities that fail to make the top K ranks among all the pairwise similarities as a function of the vector dimension N in subfigure (a) for the *full-model* and in subfigure (b) for the *abs-model*. In both cases, as mentioned in Section III-A, for assessment of robustness, the corresponding fraction for the cross full-text vs. abstract matching for the same document are also included. From the plots, we can see that for both models and the different vector dimensions explored, the model self-similarities rank among the top 4. For the *full-model*, the percentage of top self-similarities (full-full, $K = 1$ plot in Fig. 4a) and the percentage of abstract similarities for the same document that rank among the top 4 (full-abs, $K = 4$ plot in Fig. 4a) show a marked improvement with the increase in N from 100 to 300, and the improvement with further increase in N is relatively modest. For the *abs-model*, a similar, though less-pronounced trend is seen at $N = 200$. For subsequent experiments we therefore used $N = 300$ for the *full-model* and $N = 200$ for the *abs-model*. The plots in Fig. 4 also indicate that the *full-model* trained on full-text data appears to perform significantly better than the *abs-model* trained on abstracts alone: for the chosen values of the vector dimension N, over 90% of the corresponding abstracts are among the top 4 "self-similarities" for *full-model*, whereas only about 75% of the corresponding full-texts are among the top 4 "self-similarities" for *abs-model*.

Based on empirical testing, the values of the other parameters for our proposed approach were set as $\kappa = 0.35$ and $p = 10$. Table 1 lists the proportion of overlap between the top 10 recommendations obtained with the chosen parameter values and those obtained with alternative values of κ and p. We can see that the changes in the top 10 recommendations are minimal when the threshold value κ is smaller than 0.35 and $p \geq 10$. As already noted earlier, a higher p-norm favors reviewers with papers that have higher similarity scores. At $p = 10$, the top 10 recommendations already preferentially include reviewers with papers that have higher similarity scores and further increasing p has little impact. Samples of several manuscript and paper pairs indicated that papers with similarity scores below the threshold $\kappa = 0.35$ were usually not relevant to the manuscript. For κ values above 0.35, fewer papers meet the similarity threshold, resulting in fewer reviewers being recommended.

TABLE 1. Proportion of overlap between the top 10 recommendations for the chosen parameter values ($\kappa = 0.35$ and $p = 10$) and those obtained with alternative choices of the parameters κ and p.

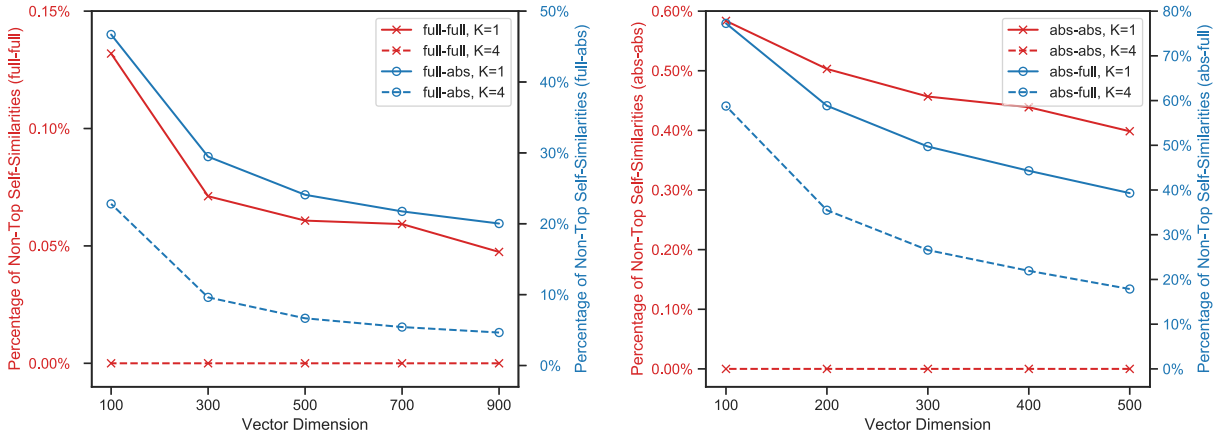
κ	Overlap Proportion	p	Overlap Proportion
0.2	0.99	1	0.66
0.25	0.99	5	0.88
0.3	0.99	15	0.99
0.4	0.71	20	0.99
0.45	0.35		
0.5	0.1		

C. PRIMARY EVALUATION ON THE IEEE ICIP 2016 DATASET

As mentioned in Section III-B, for the primary evaluation, we use the proposed approach with the *full-model* trained on the *ICIP* 2016 dataset and asked qualified judges to rate the relevance of the reviewer matches provided by our proposed approach for a random sample of manuscripts. For a rigorous evaluation of an information retrieval system, Manning *et al.* [25, Ch. 8] suggest assessing the results on at least 50 samples. Therefore, we randomly chose a subset of 75 manuscripts for our evaluation. To assess the recommendation results, we invited three judges whose combined experience and expertise included roles as technical program chair for four *ICIP*s and three editor-in-chief terms for the *IEEE Transactions on Image Processing*. The (same) subset of 75 manuscripts chosen for the evaluation was presented to the each of the judges along with the top 10 recommended reviewers[6] from the *full-model* using our proposed approach. Each judge independently provided relevance ratings on the recommendations using the methodology and the GUI described in Section III-B. In total, there were 694 recommendations to be assessed, and 2,082 relevance ratings were collected. The judges were highly appreciative of the GUI which facilitated their task. The assessments were conducted over a three week period, and each judge devoted about 20 hours to the task. Because all three judges were highly experienced and provided a valuable perspective, we used each of the individual relevance ratings instead of pooling these for the same recommendation as has been done in prior work [9]. The retrospective *ICIP* 2016 reviewer matching dataset is made publicly available [26] to facilitate further work in this area.

[5]The details are described in [24] by Mohr, an author of the Gensim library.

[6]For some manuscripts, the number of recommended reviewers was less than 10 due to the use of the cutoff κ in our matching procedure as described in Section II.

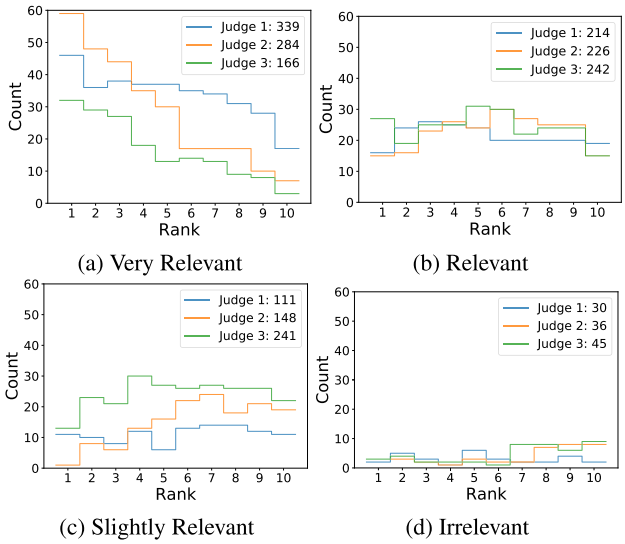


(a) *full-model* (b) *abs-model*

FIGURE 4. Model consistency test using document self-similarity rank. The fraction of self-similarities that fail to make the top K ranks are plotted as function of the vector dimension N for (a) the *full-model* trained on full text data for the full-full matching (solid line with markers) and full-abs matching (solid line) models, and (b) the *abs-model* for the abs-abs matching (solid line with markers) and abs-full matching (solid line).

TABLE 2. Statistics of judges' ratings (VR: *Very Relevant*, R: *Relevant*, SR: *Slightly Relevant*, I: *Irrelevant*) of the relevance of the reviewer matches obtained with the proposed approach.

	VR	R	SR	I
Judge 1	339	214	111	30
Judge 2	284	226	148	36
Judge 3	166	242	241	45
Total	789	682	500	111
Percentage	37.9	32.8	24.0	5.3



(a) Very Relevant (b) Relevant

(c) Slightly Relevant (d) Irrelevant

FIGURE 5. Distribution of the top ten ranked matches from our proposed approach among the four relevance rating categories for each of the judges. The total number of ratings for each judge at the relevance level are also indicated in the legend.

Table. 2 summarizes the results from our primary evaluation, where, for each judge, we list the number of recommendations assessed in each of the four relevance categories. The results indicate that the proposed system performs quite well. Collectively, 37.9% of the recommendations provided by our proposed approach are rated as *Very Relevant*; 32.8% are rated as *Relevant*; 24.0% are rated as *Slightly Relevant*; and only 5.3% were rated as *Irrelevant*. Thus 70.7% of the recommendations provided by the system were rated as relevant or very relevant, and a predominant majority of 94.7% were rated as at least slightly relevant.

Figure 5 shows how the top ten ranked matches from our proposed approach distribute among the four relevance ratings. The plots indicate that the ranks of the match scores (i.e. their relative values) do correlate with the relevance ratings; we see that the top 5 recommendations given by our framework are highly likely to be at least *Relevant*. Moreover, we see that the modes of the relevance ratings per ranking shift to the right as we move from *Very Relevant* to *Irrelevant*: the mode for *Very Relevant* is on the 1*st* rank; the mode for *Relevant* ranges from 3*rd* to 6*th*; the mode for *Slightly Relevant* ranges from 4*th* to 8*th*; and the mode for *Irrelevant* ranges from 9*th* to 10*th*. This pattern corroborates our expectation that less relevant reviewers are ranked lower. It can also be inferred that the recommendations beyond top 10 will be less relevant.

We also examined the relation between the (reviewer) match scores from our framework and the judges relevance ratings. The box-plot in Fig. 6 shows the distribution of the match scores for each of the four relevance ratings. From the plot we see that there is moderately positive correlation between the relevance ratings and the match scores, although the scores for the different relevance levels are not clearly separated from each other. This implies that while the top ranked match scores from our framework are invariably relevant,

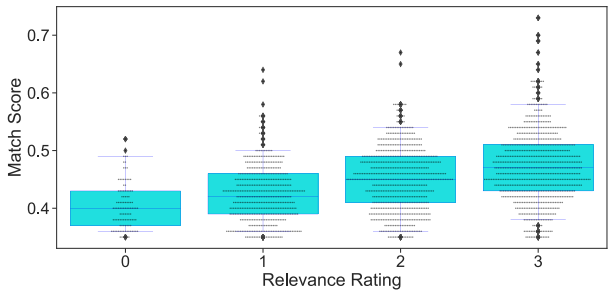


FIGURE 6. Distribution of reviewer match scores from the proposed framework for each relevance rating. The box plots show the inter-quartile range (IQR) and the whiskers identify the 5th and 95th percentiles. The actual data points are superimposed on the box plots and points with the same reviewer match score are offset to show the mass.

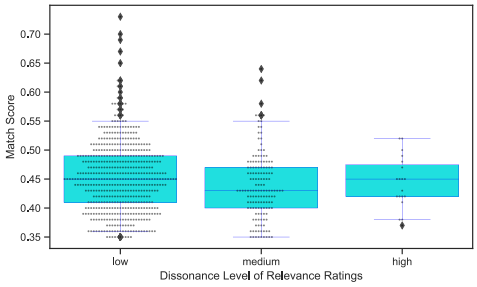


FIGURE 7. Distribution of match scores for the manuscripts as a function of the dissonance level between their relevance ratings.

by themselves, the absolute scores are not a strong predictor of relevance.

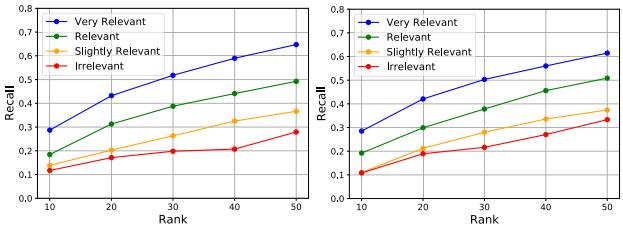
Finally, we also analyzed the level of agreement between the judges' relevance ratings. Since the relevance ratings take discrete values 0, 1, 2, 3, there are only six possible values 0, 0.47, 0.82, 0.94, 1.24, and 1.41 for the standard deviation of the judges' ratings for a given manuscript. We use a *dissonance level* to characterize the (dis)agreement between the judges and categorize a standard deviation of 0 or 0.47 into "low", 0.82 and 0.94 into "medium", 1.24 and 1.41 into "high" dissonance level. Figure 7 shows the distribution of the match scores for the low, medium, and high dissonance levels using a box-plot analogous to Fig. 6. From the plots in Fig. 7, we see that the dissonance is low for the majority of the manuscripts and only a rather small number of manuscripts are in the high dissonance bin. The dissonance level also does not seem to be correlated with the match scores from the framework, though the (few) matches corresponding to the really high scores have low or medium dissonance.

Informal feedback from the judges provided several insights. In cases where the reviewer pool included multiple authors of a paper closely matched with a manuscript, all these authors frequently all came up among the top ranked matches identified by the system. In such situations, it would clearly be desirable to have a more diverse pool of reviewers instead of allocating only authors of a single closely related paper as reviewers. While the desired diversity can be handled in subsequent subtasks that finalize the reviewer assignment based on the match scores, the observations also emphasize the need for maintaining a larger list of viable options, which could be done by taking into account how much of the different reviewers' match score arose from shared authorship papers. From an overall system perspective, these observations also highlight a key benefit of the simplicity of the proposed system: manuscripts contributing to the reviewer match score are readily identified and available, not only to explain the reason for the match to our expert judges, but also for use in subsequent tasks that need to account for other objectives beyond the expertise match. Expert judges also remarked that for some of the suggested reviewer matches for our system they were aware of conflicts of interest based on their knowledge of the research community, which were, however, not apparent in the reviewer's publications. Exclusions of such conflicts of interest, that the data may not reveal, remain problematic for our system as well as others.

D. COMPARISON AND SECONDARY EVALUATION

For comparison, we also considered two alternative options for obtaining manuscript-reviewer matches: (1) a system that computed reviewer recommendations using the *abs-model* trained using only the abstracts (for the manuscripts and the reviewers papers), and (2) an LDA model trained on full-text data with 300 topics,[7] which represents each document vector as a distribution of weights over the 300 topics allowing for reviewer recommendations to then be obtained using a procedure identical to that described in Section II, with the LDA distribution vectors replacing the *doc2vec* vectors. The performance for these alternative models was evaluated using the secondary comparison methodology described in Section III-C by computing recall at K over the same 75 manuscripts that were used in the primary evaluation and treating the judges ratings from the primary evaluation as "ground truth". The recall at 10, 20, 30, 40, and 50, for each of the four relevance categories is shown in Fig. 8 (a) and (b) for the *abs-model* and the LDA model, respectively. The plots reveal that these alternative approaches perform rather poorly; recall at 10 for both models captures only about 30% of the matches rated *Very Relevant* by the judges and only about 20% of the matches rated *Relevant* by the judges. Furthermore, these percentages increase rather slowly with increase in the rank K for the computed recall at K. For reviewer matching, the results for the larger values of K are less likely to be useful as they would also have a much larger proportion of irrelevant matches. These results highlight two important findings. First, the proposed data-driven approach for reviewer matching benefits significantly from the larger full-text training corpus compared with training on abstracts alone. This highlights the value of integrating a publisher database (to obtain full-text papers) into the approach instead of matching based on abstracts alone. Second, the performance of the LDA approach is similar to that of the

[7]The consistency check methodology of Section IV-B was also adopted for the choosing the number of topics for the LDA approach.



(a) *abs-model* (b) LDA Model

FIGURE 8. Recall at 10, 20, 30, 40, and 50 for *Very Relevant*, *Relevant*, *Slightly Relevant*, and *Irrelevant* recommendations provided by (a) the *abs-model* and (b) the LDA model.

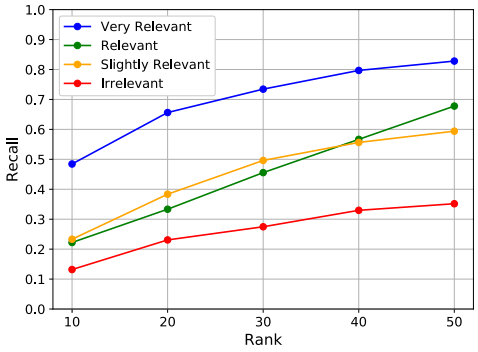


FIGURE 9. Recall at 10, 20, 30, 40, and 50 for *Very Relevant*, *Relevant*, *Slightly Relevant*, and *Irrelevant* recommendations provided by the abstract only *abs-model* in the proposed approach for the dataset in [9].

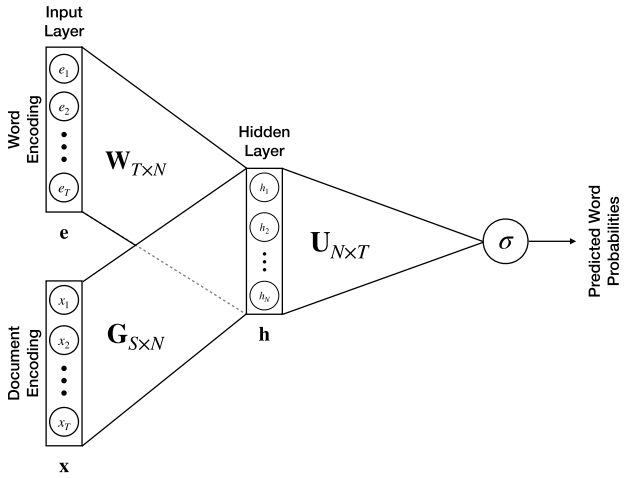


FIGURE 10. *doc2vec* model training in DBOW mode using negative sampling with simultaneous training of word vectors.

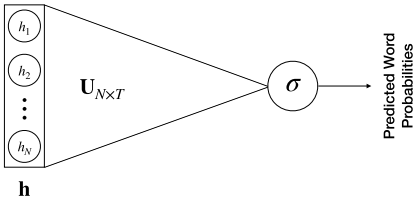


FIGURE 11. Inferring document vectors using the trained *doc2vec* DBOW model.

abs-model, which highlights the power of the learned *doc2vec* embedded vector representation when trained over the larger full-text database compared to LDA's more traditional topic based modeling framework.

Finally, we also performed a limited secondary evaluation of our proposed framework over the *NIPS* 2006 dataset from Mimno & McCallum [9]. Described in our terminology, the dataset contains 148 manuscript abstracts, a list of 364 prospective reviewers and abstracts of their papers. Expert judges' ratings for reviewer match recommendations provided by the approach in [9] for 34 manuscripts are also included in the dataset using the four-level relevance rating that we also adopted in this paper and described in Section III-B. Specifically 393 ratings of relevance for manuscript-reviewer matches are provided in the dataset, which were obtained by aggregating input from nine judges and pooling the rating for each match by taking the minimum value over the different relevance ratings provided by the judges. For our proposed methodology, we trained the *abs-model* on the abstracts, which is the only option because full text papers are unavailable in this dataset. We then evaluated the performance for our trained *abs-model* using the secondary comparison methodology described in Section III-C by computing recall at K over the 34 manuscripts, once again treating the pooled judges ratings from the primary evaluation as "ground truth". Figure 9 shows a plot of the recall at 10, 20, 30, 40, and 50, for each of the four relevance categories for the *abs-model* obtained using our proposed methodology. Even though the *abs-model* is handicapped by the lack of the richer full-text data, the performance is significantly better than what was seen in Fig. 8 (a). Almost 50% of the very relevant recommendations and over 20% of the relevant and slightly relevant references are included within the top 10 ranked recommendations from the *abs-model* and, compared with Fig. 8 (a), these percentages also increase faster with increase in the rank K for the computed recall at K.

V. LIMITATIONS OF THE STUDY

Objective evaluation and comparison of reviewer matching systems faces a fundamental challenge due to the fact that there is no unique ground truth for reviewer matching (as already mentioned earlier). Unlike traditional pattern classification problems, reviewer to manuscript matches are inherently multifaceted and nonunique. For a manuscript and reviewer corpus, if assessments of the suitability of each possible manuscript-reviewer pairing were available from qualified judges, one could formulate objective metrics for evaluation and comparison of reviewer matching systems. However, the time commitment required to perform such an exhaustive pairwise assessment and the associated tedium make it untenable. For this reason, for our primary evaluation, we relied on judges assessment of the recommendations

provided by our system, which is also the approach adopted in past studies on reviewer matching. We also propose alternative analyses in this paper that partly overcome the limitation. Specifically, the consistency test of Section III-A allows us to make parameter choices for our data driven models without requiring extensive manual input and the secondary evaluation of Section III-C allows us to leverage prior evaluation datasets for limited comparison.

Our work in this paper focused solely on the subtask of expertise matching using the affinity between the reviewers' prior papers and the manuscripts. While this affinity matching is a crucial ingredient, reviewer assignment, in its entirety, is a complex problem that must address several additional subtasks. Important additional considerations include, for example, reviewer coverage of different technical areas that contribute to a manuscript's innovation, diversity of reviewer affiliation, workload balance/restrictions from reviewers, concurrency of the reviewers' research interests with the topic of the manuscript, and accounting for conflicts of interest and ensuring fairness of assessment. The approach presented in this paper can be integrated within the larger reviewer assignment task by utilizing the manuscript-reviewer match scores in a subsequent manual reviewer allocation phase that is aligned with traditional practices, or in an automated system that seeks to automate more of the subtasks involved in reviewer assignment. The simplicity of the proposed approach and linkages to publicly accessible versions of the authors' papers on a publisher's website lends itself to better explainability and exploration of the data underlying the match scores, which facilitated the expert judges' relevance ratings and can also be helpful in downstream subtasks for reviewer allocation. Our approach used minimal conflict of interest filtering, and, during the primary evaluation, the judges indicated that they saw some recommendations with conflicts of interest. These could be mitigated by using both better identity management such as ORCIDs [27] for authors, if available, and co-author lists from reviewers' papers or other sources such as academic authorship graphs [28]. We also note that a key limitation of a system of expertise matching based on published papers is that it tends to exclude industry practitioners who may have deep and directly relevant experience and expertise but may not be actively involved in publishing formal papers. In this regard, approaches that can meaningfully also harness less formal publication venues such as StackExchange [29] would be of interest.

VI. CONCLUSION

The framework that we developed and evaluated in this paper provides a effective, automated, data-science based approach for finding reviewers with expertise that is matched with manuscripts submitted for review. Of the recommendations provided by our framework, expert judges rated over 70% as *Relevant* or better, an overwhelming 95% as at least *Slightly Relevant*, and under 5% as *Irrelevant*. The approach relies on a relatively simple data-driven methodology using finite dimensional vector space embeddings for documents, where the embeddings are learned from a corpus of reviewers' publications. The methodology for evaluation and extensive tests at real-world scale also constitute key contributions of our work: combining an automated consistency check that allows parameter setting, a user interface for easing the judges' burden of providing relevance ratings, and a secondary indirect evaluation approach that allows comparisons against alternative options and techniques and also on other datasets. Our primary evaluation was performed using full-text documents where the reviewers' papers were obtained directly from a publisher database, which allowed evaluation in a realistic large-scale setting. Our experiments and evaluation also revealed that rich full-text data significantly improves the effectiveness of the approach: recommendations from the *full-model* trained on full-text data perform significantly better than the *abs-model* that was trained only on abstracts. With the rich full-text data, our data-driven approach also outperformed a more traditional LDA topic modeling implementation.

APPENDIX

DOC2VEC BACKGROUND

Both *doc2vec* and its predecessor *word2vec* are based on a fully-connected two-layer neural network architecture that is trained with the objective of predicting which words occur in a pre-defined context in the training document corpus. There are several variants of these models that differ in the context they use, the objective function they seek to optimize, and the approximations and heuristic simplifications they exploit to speed up training [13], [30]. We focus our description on the efficient negative-sampling based DBOW version of *doc2vec* with simultaneously trained word vectors. As indicated in Section IV-B, we chose this specific version of *doc2vec* for use in our reviewer recommendation task based on prior reported benchmarks and guidance [14], [18]. Figure 10 schematically illustrates the two-layer structure used for the model training. The learned parameters of the model comprise three matrices $\mathbf{W}_{T\times N}$, $\mathbf{G}_{S\times N}$, and $\mathbf{U}_{N\times T}$ where T denotes the vocabulary size, i.e., the number of distinct words in the document corpus, S denotes the number of documents in the training corpus, and N is the model hyperparameter denoting the size of the word/document vectors. Instead of formulating a well-defined posterior distribution in terms of these three matrices, the negative sampling version of *doc2vec* (and *word2vec*), instead makes use of a simplified training procedure that offers very significant computational savings.[8] A document (or word) is sampled from the training corpus and a corresponding hidden layer vector $\mathbf{h}$ is obtained as $\mathbf{h} = \mathbf{G}^\top\mathbf{x}$ (or $\mathbf{h} = \mathbf{W}^\top\mathbf{e}$), where $\mathbf{e}$ (or $\mathbf{x}$) is the one-hot encoded representation of the document (or word), i.e., a vector in which the entry corresponding to the index of the document in the corpus (or word in the vocabulary)

[8] In the word2vec setting, this simplified training procedure is derived and justified as an approximation in [31].

is 1 and all other entries are 0. A (positive) word from the document encoded by $\mathbf{x}$ (or from the context window of the word encoded by $\mathbf{e}$) is sampled and encoded as its index w_o in the vocabulary. Additionally, K negative sample words, i.e., words not in the document encoded by $\mathbf{x}$ (or not in the context window of the word encoded by $\mathbf{e}$) are sampled according to a "noise distribution" over the vocabulary.[9] Through back-propagation, the relevant entries in the matrices $\mathbf{G}_{S\times N}$ (or $\mathbf{W}_{T\times N}$) and $\mathbf{U}_{N\times T}$ are then updated via a gradient descent update that seeks to minimize the objective function

$$-\log\sigma\left(\mathbf{U}_{w_o}^{\top}\mathbf{h}\right)-\sum_{w\in\mathcal{W}_{\text{neg}}}\log\sigma\left(-\mathbf{U}_{w}^{\top}\mathbf{h}\right), \tag{5}$$

where $\mathcal{W}_{\text{neg}}$ is the set of negative word samples, and

$$\sigma(u)=\frac{1}{1+e^{-u}}, \tag{6}$$

denotes the logistic function. The process is then repeated for another sampled word and corresponding negative samples. The overall training process iterates between using input document or word samples, where the former involves updates of $\mathbf{G}_{S\times N}$ and $\mathbf{U}_{N\times T}$ and the latter those of $\mathbf{W}_{T\times N}$ and $\mathbf{U}_{N\times T}$. Once training is completed, an inference process estimates a document vector $\boldsymbol{v}$ corresponding to a given document using the matrix $\mathbf{U}_{N\times T}$ as illustrated schematically in Fig. 11. Specifically, starting with a random initialization, back-propagation is used to iteratively update the vector $\mathbf{h}$, once using the objective function (5) where $\mathbf{U}_{N\times T}$ is fixed and the true and negative samples are for the given document. The final version of $\mathbf{h}$ obtained from this update process is the document vector $\mathbf{v}$ for the given document.

ACKNOWLEDGMENT

This work was done while Yue Zhao was at the University of Rochester. We thank the expert judges for providing relevance ratings for the manuscript-reviewer recommendations and Manuel Rechani and Prakash Bellur from the IEEE Xplore platform management team for support with the IEEE Xplore API and SDK and for supplying us PDFs of the full text papers for our reviewer corpus. We also thank the Center for Integrated Research Computing (CIRC), University of Rochester, for providing computational resources for this work and Harry Stern and CIRC for technical support for the project. Yue Zhao would like to thank his employer, the Rochester Data Science Consortium, for supporting his involvement in the project.

REFERENCES

[1] S. Price and P. A. Flach, "Computational support for academic peer review: A perspective from artificial intelligence," *Commun. ACM*, vol. 60, no. 3, pp. 70–79, Mar. 2017, doi: 10.1145/2979672.

[2] Y.-R. Lin, H. Tong, J. Tang, and K. S. Candan, "Guest editorial: Big scholar data discovery and collaboration," *IEEE Trans. Big Data*, vol. 2, no. 1, pp. 1–2, Mar. 2016, doi: 10.1109/TBDATA.2016.2562840.

[9]The default option of a unigram distribution raised to the 3/4-th power is utilized as the noise distribution, which was empirically shown in [17] to provide the best results.

[3] S. Hettich and M. J. Pazzani, "Mining for proposal reviewers: Lessons learned at the national science foundation," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2006, pp. 862–871, doi: 10.1145/1150402.1150521.

[4] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, New York, NY, USA: McGraw-Hill, 1983.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[6] L. Charlin and R. Zemel, "The Toronto paper matching system: An automated paper-reviewer assignment system," in *Proc. ICML Workshop Peer Reviewing Publishing Models (PEER)*, 2013, pp. 1–9, [Online]. Available: https://openreview.net/forum?id=caynafZAnBafx

[7] B. Li and Y. T. Hou, "The new automated IEEE INFOCOM review assignment system," *IEEE Netw.*, vol. 30, no. 5, pp. 18–24, Sep. 2016, doi: 10.1109/MNET.2016.7579022.

[8] M. Karimzadehgan, C. Zhai, and G. Belford, "Multi-aspect expertise matching for review assignment," in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*. New York, NY, USA: ACM, 2008, pp. 1113–1122, doi: 10.1145/1458082.1458230.

[9] D. Mimno and A. McCallum, "Expertise modeling for matching papers with reviewers," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2007, pp. 500–509, doi: 10.1145/1281192.1281247.

[10] C. Ferguson, A. Marcus, and I. Oransky, "Publishing: The peer-review scam," *Nature*, vol. 515, no. 7528, pp. 480–482, Nov. 2014, doi: 10.1038/515480a.

[11] E. Callaway, "Faked peer reviews prompt 64 retractions," *Nature*, vol. 785, pp. 23–25, Aug. 2015, doi: 10.1038/nature.2015.18202.

[12] H. Rivera, "Fake peer review and inappropriate authorship are real evils," *J. Korean Med. Sci.*, vol. 34, no. 2, p e6, 2019, doi: 10.3346/jkms.2019.34.e6.

[13] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[14] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," 2015, *arXiv:1507.07998*.

[15] G. Soğancioğlu, H. Öztürk, and A. Özgür, "BIOSSES: A semantic sentence similarity estimation system for the biomedical domain," *Bioinformatics*, vol. 33, no. 14, pp. i49–i58, Jul. 2017.

[16] M.-S. Duma and W. Menzel, "SEF@UHH at SemEval-2017 task 1: Unsupervised knowledge-free semantic textual similarity via paragraph vector," in *Proc. 11th Int. Workshop Semantic Eval.*, Vancouver, BC, Canada, Aug. 2017, pp. 170–174, doi: 10.18653/v1/S17-2024.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[18] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," in *Proc. 1st Workshop Represent. Learn. NLP*. Berlin, Germany: ACL, Aug. 2016, pp. 78–86, doi: 10.18653/v1/W16-1609.

[19] Google Groups. *Handle Error of Unseen Words in Doc2vec*. Accessed: Mar. 2020. [Online]. Available: https://groups.google.com/forum/#!topic/gensim/Ucc9JogRc-4

[20] IEEE. *IEEE Xplore Digital Library*. Accessed: Mar. 2020. [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp

[21] IEEE. *Software Development Kit*. Accessed: Mar. 2020. [Online]. Available: https://developer.ieee.org/Python Software Development Kit

[22] XpdfReader. *Pdftotext*. Accessed: Mar. 2020. [Online]. Available: https://www.xpdfreader.com/pdftotext-man.html

[23] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*, Valletta, Malta, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

[24] Google Groups. *Doc2Vec-Diff Between PV-DM & PV-DBOW?*. Accessed: Mar. 2020. [Online]. Available: https://groups.google.com/forum/#!topic/gensim/uC6I47JtIps

[25] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.

[26] Y. Zhao, A. Anand, and G. Sharma, *RetroRevMatchEvalICIP16: A Retrospective Reviewer Matching Dataset and Evaluation for IEEE ICIP 2016*, IEEE Dataport, 2021, doi: 10.21227/ez82-ez41.

[27] *Open Researcher and Contributor ID*. Accessed: May 2021. [Online]. Available: https://orcid.org/

[28] *Microsoft Academic Graph*. Accessed: May 2021. [Online]. Available: https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

[29] *StackExchange*. Accessed: May 2021. [Online]. Available: https://stackexchange.com/
[30] X. Rong, "Word2vec parameter learning explained," 2014, *arXiv:1411.2738*.
[31] Y. Goldberg and O. Levy, "Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*.

YUE ZHAO received the B.A. degree in applied mathematics (with a focus on civil engineering) from the University of Wisconsin-Madison, in 2016, and the M.S. degree in data science from the University of Rochester, in 2018. He has been working as the Principal Investigator of natural language processing and text mining projects at the Rochester Data Science Consortium, since 2019.

AJAY ANAND received the M.S. degree in biomedical engineering from the University of Texas Southwestern Medical Center, Dallas, TX, USA, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, WA, USA, in 2003 and 2005, respectively. Prior to joining the University of Rochester, he was with Philips Research North America, Briarcliff Manor, NY, USA, from 2005 to 2015, as a Senior Research Scientist and a Technical Project Leader at the Medical Ultrasound Imaging and Therapy Group, and then with Carestream Health, from 2015 to 2017, as a Technical Project Manager for their commercial ultrasound imaging platform. He has been an Associate Professor and the Deputy Director of the Goergen Institute for Data Science, University of Rochester, since 2017. He serves as a principal investigator on a multi-year NSF-funded training program aimed at expanding research opportunities for undergraduate students in interdisciplinary areas within data science. His research interests include healthcare data analytics, biomedical signal processing, and biomedical instrumentation.

GAURAV SHARMA (Fellow, IEEE) received the B.E. degree in electronics and communication engineering from the Indian Institute of Technology, Roorkee (formerly, University of Roorkee), the master's degree in applied mathematics from North Carolina State University (NCSU), Raleigh, NC, USA, and electrical communication engineering from the Indian Institute of Science, Bengaluru, India, and the Ph.D. degree in electrical and computer engineering from NCSU. From 1996 to 2003, he was with Xerox Research and Technology, Webster, NY, USA, first as a member of research and technology staff and then as a Principal Scientist and a Project Leader. From 2008 to 2010, he was the Director of the Center for Emerging and Innovative Sciences (CEIS), a New York state supported center for promoting joint university-industry research and technology development, which is housed at the University of Rochester. He is currently with the Department of Electrical and Computer Engineering, the Department of Computer Science, and the Department of Biostatistics and Computational Biology, University of Rochester. His research interests include data analytics, signal and image processing, computer vision, color imaging, media security, and communications. He is a fellow of SPIE and the Society of Imaging Science and Technology (IS&T). He has served as the Editor-in-Chief (EIC) for the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), from 2018 to 2020, and the *Journal of Electronic Imaging* (JEI), from 2011 to 2015.