Is the Paper Within Scope? Are You Fishing in the Right Pond?

Addressing the Appropriateness of a Manuscript to a Journal in the Peer Review Workflow

Tirthankar Ghosal, Ravi Sonam, Asif Ekbal, Sriparna Saha, Pushpak Bhattacharyya
(tirthankar.pcs16,ravi.cs13,asif,sriparna,pb)@iitp.ac.in

ABSTRACT

Outright rejection from the editors' desk, better known as pre-screening or desk-rejection is an unfortunate yet common occurrence in academic peer review. In spite of having merit, many papers are rejected from the desk merely because they are a misfit to the scope of the journal. However, this phenomena costs a considerable time of both the editors and the authors. In this work, we present an investigation towards automation of desk rejection for *out-of-scope* submissions. We model the problem as a binary classification decision of an article being within scope or outside. We carry our experiments on six different Elsevier Computer Science journals. Our approach based on supervised machine learning outperforms a state-of-the-art by a wide margin in terms of accuracy (at least ~8%). We believe that our proposed method is generic, and with requisite set-up could be applied to articles of other journals. An appropriate system developed with our features could also help prospective authors to check beforehand whether they are submitting to the right venue.

KEYWORDS

desk rejection, scope detection, content analysis, scholarly texts, bibliographic analysis

1 INTRODUCTION

Rejection is the norm in academic peer review, and desk rejection is one such woe faced by most scholars during their career. The first step in the peer review process is the initial screening, usually performed by the editors, where they decide whether a prospective scholarly article should be rejected without further review or forwarded to expert reviewers for meticulous evaluation. With the exponential growth in scholarly communications, it is increasingly becoming difficult for the editors to manually go through each submission and respond to the authors in a reasonable time. Observations from editorial communications [12] and statistics [5] reveal that a major cause (~30%) for desk rejection is that the article is not within the scope of the journal to which it was sent. A good amount of precious time of both authors and editors gets wasted in the process. The author may have considered some other venue and the editors would not have been burdened with the massive load of irrelevant submissions. This work of ours is an attempt to mitigate this seemingly time-consuming problem and provide a machine learning solution to it. We strive to seek automation that would benefit both the scholars and editors to judge the appropriateness of a specific scientific article to the scope of the prospective journal and thereby assist them in making intuitive decisions. We extract features from almost every section of a scientific manuscript that could contribute to identifying its domain: *Author, Content and Bibliography*. Our point of departure for this particular work is the bibliography section of research articles. We hypothesise that with obvious exceptions *if an article belongs to a particular domain then the majority of its references would fall in that certain domain.* Coupled with other factors, our approach *ScopeJr* achieves *state-of-the-art* performance across six different journals. However, we agree that this preliminary work may not hold universally for all journals as the nature of the scope of different journals is different. The current work is a consequence of our ongoing effort towards an AI-assisted peer review system.

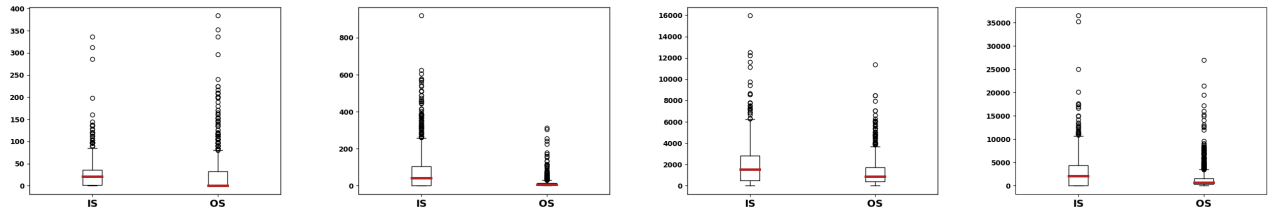
2 SCOPE DETECTION

We frame our investigation as a binary classification problem of research articles (IS: *in-scope* and OS: *out-of-scope* classes). Given a journal $\mathcal{J}$ and a paper P, we seek to answer: *If P is within the scope of $\mathcal{J}$.* Articles already published signify that they are within-the-scope and somewhat define the *domain-of-operation* of a journal. Our *out-of-scope* data are those desk-rejected manuscripts which according to the editors are not a good fit to the topical coverage and aspirations of the journal. To the best of our knowledge, this work is the first attempt towards automatic scope detection for peer review. Journal publishers usually have their own recommender systems[1] which mostly rely on domain-specific vocabulary match between the prospective article and the accepted articles of different journals to throw a suggestion. A few works in literature are eTBLAST[11], Jane[4], and certain journal recommender systems as explored in [1–3, 13]. Our findings suggest that there are more aspects to explore than merely vocabulary match to judge the suitability of the submitted article. We achieve significant improvement over the state-of-the-art[6]. Our approach furnishes the potential in the design of a system that could help the editors in identifying potential misfit submissions easily. However, we agree that a binary decision will not be a welcome solution and we intend to extend this *work-in-progress* to design a classification system with a confidence score and feedback loop; also investigate the feasibility of a recommender system of academic journals with our feature set.

3 DATA DESCRIPTION AND PREPROCESSING

We consider all accepted (ACC) articles from six different Elsevier Computer Science journals: Artificial Intelligence (ARTINT), Computer Networks (COMNET), Statistics and Probability Letters (STATPRO), Theoretical Computer Science (TCS), Computer Standards and Interfaces (CSI), and Simulation Modeling Practice and Theory (SIMPAT), to build our domain lists (Section 4.1). However, for our experiments, we use 1000 exclusive accepted articles from each journal as our *In-Scope* data. We procure and curate 1000 *out-of-scope* articles for each of these journals internally from Elsevier (most of them were actually desk-rejected due to out-of-scope, and some were accepted articles of distantly related journals[2]). After a thorough study of these data, we come up with some important

[1] for, e.g. http://journalfinder.elsevier.com/
[2] Rejected articles are confidential and hard to get



(a) Keyword overlap with past ACC data (STATPRO) **(b) Referenced paper overlap with past ACC data (ARTINT)** **(c) Referenced venue overlap with past ACC data (COMNET)** **(d) Author overlap with past ACC data (ARTINT)**

Figure 1: Box plots of various factors across an exclusive set of 1000 IS and 1000 OS articles. The match is in terms of overlap of keywords, referenced paper titles, bibliographic venues and authors with respect to past accepted papers of each journal. The median is always high for IS w.r.t. OS articles.

observations (Figure 1). There are noticeable differences among In-Scope (IS) and Out-of-Scope (OS) articles in terms of the keywords they use, bibliography (papers and venues) they refer and their authors when compared to corresponding past accepted papers. For e.g., the median of bibliographic title overlap for IS articles against ARTINT-ACC articles is found to be 43 whereas the same for OS articles is 7 (Figure 1(b)). We observe similar contrast in data distribution across IS and OS articles for various other factors (venue, keyword, authors) as well (Figure 1). Hence we curate our features (Section 4.2) based on these observations. We parse[3] the scientific articles, originally in .pdf format, to generate a corresponding .xml document consisting of essential information within structured XML tags. We then extract the following information from these .xml versions: *Title, Author names, Abstract, Author-listed keywords, Body-text, Bibliographic Paper Titles and corresponding Venues.* The extracted data are noisy, and we perform certain **pre-processing**:
(1) Removed editions from conference names and mapped different editions of the same conference and abbreviations into one. For e.g., *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking* →ACM International Conference on Computer and Communications Security→CCS
(2) Mapped variants of certain words in conference or journal names via regular expressions. For e.g. *Jour.* →*Journal*, *Trans.* →*Transactions*, *Distrib.* →*Distributed*

4 METHODOLOGY

4.1 Building Domain Lists

As the past accepted articles are strong indicators of the *domain of operation,* or *scope* of a particular journal, we build our features based on the information extracted from those published ones. We pay special attention to the *bibliography* section. For each journal, we create several exhaustive lists:
(L1) A **Keyword Dictionary** consisting of *author-listed* keywords and record their frequency of occurrences (average 5k+ keywords per journal)
(L2) **Bibliographic Title List** (average 30k+ titles per journal)
(L3) **Bibliographic Venue List** (average 7k+ venues per journal)
We hypothesise that for a particular journal, *the relative importance of some papers and certain venues would always be high if measured across all published articles.* Those certain papers/venues are the representative data points of that journal. Hence we extract all referenced paper titles and corresponding venues from the bibliography section of ACC articles. For each entry X in **L2, L3**:

$$V(X) = \sum_{j=1}^{n} CitE(X) \tag{1}$$

where X could either be a **paper title** or a **venue** (journal or meeting/conference/workshop) in the reference section of article j. n is the number of ACC articles. We define a novel function **Citation Effect** (*CitE*) which:

- corresponds to the number of in-citations of X within the body of a candidate article j **if** X **is a paper title.**
- corresponds to the number of occurrences of X within the bibliography section of article j **if** X **is a venue**.

(L4) **Author List** (average 15k+ authors per journal)
The intuitions behind creating such lists are:
(1) Articles which are highly in-cited within a particular journal have higher relevance to the scope of the journal.
(2) Similarly, venues which have a higher presence in the bibliography section of a particular journal, are of higher relevance to the scope of that journal.
(3) More an author publishes articles belonging to a certain domain; greater is the chance that her prospective next would belong to the same domain (authors' favourite). We record the publication frequency of authors in each journal separately.

4.2 Feature Engineering

(a) **Weighted Keyword Match[wt_kw_m]**: We design this feature to emphasise the containment and relative importance of the keywords in the candidate article with respect to the Keyword Dictionary. The value for this feature for a candidate article Y is:

$$KWScore_Y = \frac{|KW_Y \cap KW_D|}{|KW_Y|} \times \sum_{i=1}^{|KW_Y \cap KW_D|} f(K_i)$$

where KW_Y is the set of author-defined keywords in the candidate article Y, KW_D is the set of keywords in the Keyword Dictionary D, $f(K_i)$ is the frequency of keyword K_i as listed in D, and $K_i \in \{KW_Y \cap KW_D\}$. Frequently occurring keywords are domain-specific words, hence have higher weights.
(b)**Title Scope** and **Venue Scope**: We calculate these features from the bibliography section of a candidate article Y. From the two exhaustive lists of paper titles and venues, we calculate the Title

[3]using GROBID: https://github.com/kermitt2/grobid

Scope (T_Y) and Venue Scope (V_Y) respectively:

$$T_Y = \sum_{k=1}^{m}[V(t_k) * CitE(t_k)] \qquad V_Y = \sum_{k=1}^{m} V(v_k)$$

where m is the total number of bibliographical references in Y ; $V(t_k)$ is derived from table look-up Bib-Title List (Eq.1) and $CitE(t_k)$ is the citation effect of k-th title in Y. Similarly $V(v_k)$ is derived from table look-up Bib-Venue List.[**bib_tit_sc, bib_jr_sc, bib_cnf_sc**]

(c) **Author Domain Publication Frequency**[**adpf**] For a candidate article, we take the summation of the publication frequency of its authors in the concerned journal from the author list.

(d) **Distance From Cluster of Similar Articles**[**clust_dist**] The accepted articles of a specific journal are grouped into clusters representing different sub-domains within the journal scope. Thus the distance of a given research article from the set of clusters formed on the accepted articles may contribute to determine its scope. Any outlier to such clusters may be considered as *out-of-scope*. With this intuition we perform the steps in Algorithm 1. For each journal, we generate clusters from all accepted articles. We then take *minimum* of the distances of the candidate article Y from the cluster centers, in order to learn how close is Y to any of the clusters so formed.

Algorithm 1 Calculate distance from journal cluster boundary

1. Use RAKE[10] to automatically extract keywords from the Title, Abstract, Introduction and Conclusion sections of an article Y belonging to journal J.
2. Use *word2vec*[9] to generate the vectors of the extracted keywords (top 30 ranked RAKE extracted keywords) from Y.
3. Calculate the document vector of Y by concatenating all the keyword vectors from *Step 2*.
4. Repeat *Steps 1-3* for all the accepted articles of the journal J.
5. Use Word Mover's Distance (WMD)[8] as the distance metric between two document vectors and generate the similarity matrix.
6. Apply K-Medoids[7] on the similarity matrix from *Step 4* to generate the clusters (C_i) [K is determined via Silhouette Index;user tune-able;can vary across journals]
7. Find the radius(r_i) of a cluster C_i as:
$$r_i = median(distance(c_i, p_j))$$
where c_i is the centre of cluster C_i and p_j is any point within cluster C_i.
8. Find the document vector (p_Y) of a candidate article Y using *Steps 1-3*.
9. Distance of the candidate article Y from the boundary of cluster C_i is given as :
$$D_i = distance(c_i, p_Y) - r_i$$
10. Repeat *Step 9* for all the clusters (C_i) obtained from *Step 6* to get :
$$D_Y = minimum(D_i)$$

Computer Science Specific Word Embeddings. *One major contribution in executing this feature is the creation and usage of* ***word2vec****[9] word vectors trained on the entire Computer Science journal articles of Elsevier (to preserve domain dependency).* We processed 41737169 sentences from around 400K articles. The embedding dimension is set to 300. We choose lines of texts extracted from *Title, Abstract, Introduction, Body, Conclusions* sections of accepted articles. Certain preprocessing needs are: removal of special characters, headings, table and figure captions, etc.

5 EVALUATION

To evaluate the performance of our system we employ a range of classifiers[4] on our feature set. However due to the inter-dependent nature of our features we find that Random Forest performs the best across all journals. We coin our approach using Random Forest classifier as ***ScopeJr***. For each of the journals we take 1000 exclusive accepted papers as *in-scope* data and 1000 *out-of-scope* articles as rejected data. We extract features and perform the experiments in a *10-fold cross-validation* classification set up. Finally we compare the classification performance of our proposed system with the *state-of-the-art* **Elsevier Journal Finder (EJF)**[6] on the same dataset and report the results (Table 1). EJF is a *state-of-the-art* recommender system provided by Elsevier solutions to the academic fraternity which recommends highly relevant journals to the authors for their papers. Elsevier Journal Finder takes as input the *Title* and *Abstract* of a prospective scientific article (Y) and presents a list of 10 relevant Elsevier journals (J) to the user as output which s/he may consider for submitting her/his article. Although the recommended journals are limited only to Elsevier published ones, but it is to be noted that Elsevier has more than 2900 peer-reviewed journals that cover almost all major scientific domains. Although we had *true class* labels from Elsevier data, we follow heuristics to determine the **EJF** *predicted* class label of a prospective article Y : *If EJF suggests J for* $Y \rightarrow Y$ *is* ***In-Scope*** *of* $\mathcal{J}$ *otherwise,* ***EJF*** *deems* Y *to be* ***Out-of-Scope*** *for* $\mathcal{J}$.

Baseline: We take the weighted overlap of keywords extracted from *Title*, *Abstract* with Keyword Dictionary (D) as features. We use standard Support Vector Machine (SVM) as the classifier.

6 RESULTS AND OBSERVATIONS

Results reported in Table 1 demonstrate the richness of our feature set. Using our feature set with Random Forest (RF), our approach *ScopeJr* performs way better than the baseline and Elsevier Journal Finder (EJF). Except for in SIMPAT, we achieve an improvement of over 20% in terms of accuracy. The comparatively low performance in SIMPAT is because SIMPAT has a wider scope, mostly simulations of different theories, and accepts articles from different disciplines. Since we are particularly interested in a pruning perspective, we report *out-of-scope* (OS) results. Thorough analysis of data and experimental results led us to the following observations:

(1) *Bibliographic* features have induced significant improvements (Figure 2) because we deduce the *Bibliographic feature* values from within the body section of the scientific articles. *When a certain portion of a scientific article cites a reference, the scope of that portion is influenced by the domain of referenced article. The domain of the cited reference exerts local influence on that portion of the scientific article.* So if many in-domain references are cited in distributed portions of a research article, quite possibly the entire research article falls in the same domain. We measure *in-domain* or *in-scope* by simply counting occurrences across published articles of a certain journal; higher the better.

(2) For all the journals our approach outperforms the **EJF** in terms

[4] using the popular machine learning toolkit WEKA

Journals→	ARTINT			COMNET			STATPRO			TCS			CSI			SIMPAT		
App.↓	P	R	A	P	R	A	P	R	A	P	R	A	P	R	A	P	R	A
Tit.+Abs. ‡	0.49	0.58	55.8	0.56	0.64	58.2	0.44	0.49	48.9	0.43	0.45	46.4	0.49	0.58	61.2	0.54	0.63	63.2
EJF	0.54	0.62	63.6	0.34	0.43	44.4	0.43	0.52	53.5	0.55	0.64	66.8	0.51	0.67	65.6	0.53	0.65	64.8
ScopeJr	0.89	0.86	**87.2**†	0.82	0.80	**81.4**†	0.83	0.84	**83.9**†	0.86	0.87	**87.2**†	0.81	0.95	**86.7** †	0.72	0.76	**72.2**†

Table 1: Scope-Check figures for *out-of-scope* (OS) class across 6 journals, $P \rightarrow$ *Precision*, $R \rightarrow$ *Recall*, App. → Approaches, ‡ → Baseline using only Title (Tit.) and Abstract (Abs.) with SVM classifier. The Accuracy values (†) for *ScopeJr* are statistically significant over EJF performance (two-tailed t-test, p<0.05)

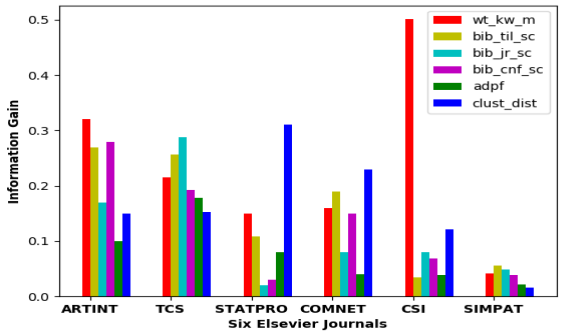


Figure 2: Significance of features observed by ranking features based on Information Gain

of precision, recall and accuracy values. This is due to the fact that EJF considers only the *Title* and *Abstract* sections of a research article and uses the **Elsevier Finger Print Engine**[5] based on identification of *Noun Phrases* from those sections. Our method goes beyond this idea and we use *Bibliographic, Author and Content* information which highly contributes the towards categorization.
(3) Some journal specific features (like presence of mathematical expressions for STATPRO) may further improvise performance.
(4) For journals having very wider scope (for e.g., Computer Science Review or Nature or Science) or multi-disciplinary in nature, this approach may not be fruitful.
(5) Scope of a journal gets more compact and streamlined with time. Hence experimenting with only recent articles instead of historical ones may boost the performance.
(6) Journals SIMPAT and CSI accept papers across many domains. Hence we observe information from several domains in their *Bibliography* section. However for ARTINT, STATPRO, and COMNET we find *Bibliography* generates a comparatively restricted *domain-specific* set and hence bibliographic features proved more effective.
(7) Some authors co-author multiple publications in the same journal which signifies their area of interests. New authors usually have supervisors as co-authors; hence we do a summation of the frequency of their publications. We see *adpf* feature has less significance in comparison to others. However, this feature could be important if we consider an entire domain consisting of different journals as the reference list.

7 CONCLUSION AND FUTURE WORKS

Our work comes upon with important insights into determination of *scope* of a scientific article. We provide ample empirical pieces of evidence to justify our claim that if we look beyond Title and Abstract, we may get a more detailed understanding of the scope of a prospective manuscript. We believe our approach is generic and with obvious exceptions could be adapted to other journals. Our proposed system could aid in identifying a large number of *out-of-scope* articles that reach the editors' desk assist her to make quick decisions, and eventually speed-up the overall peer review process. Next, we would investigate deep multimodal learning to extract features from domain-related journals as well as journals having a decidedly broader scope. We would also like to investigate how a model trained on one journal performs on another within the same domain to address the cold-start problem for new journals.

[5]https://www.elsevier.com/solutions/elsevier-fingerprint-engine

ACKNOWLEDGMENTS

The first and third author gratefully acknowledge Visvesvaraya PhD Scheme and YFRF under Ministry of Electronics and Information Technology (MeitY), Government of India for supporting this research. We also extend our gratitude to Elsevier for data support.

REFERENCES

[1] Hamed Alhoori and Richard Furuta. 2017. Recommendation of scholarly venues based on dynamic user interests. *Journal of Informetrics* 11, 2 (2017), 553–563.
[2] Imen Boukhris and Raouia Ayachi. 2014. A novel personalized academic venue hybrid recommender. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*. IEEE, 465–470.
[3] Zhen Chen, Feng Xia, Huizhen Jiang, Haifeng Liu, and Jun Zhang. 2015. AVER: Random walk based academic venue recommendation. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 579–584.
[4] Mounir Errami, Jonathan D Wren, Justin M Hicks, and Harold R Garner. 2007. eTBLAST: a web server to identify expert reviewers, appropriate journals and similar publications. *Nucleic acids research* 35, suppl 2 (2007), W12–W15.
[5] Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2018. Investigating Impact Features in Editorial Pre-Screening of Research Papers. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*. 333–334. https://doi.org/10.1145/3197026.3203910
[6] Ning Kang, Marius A Doornenbal, and Robert JA Schijvenaars. 2015. Elsevier journal finder: recommending journals for your paper. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 261–264.
[7] Leonard Kaufman and Peter J Rousseeuw. 1990. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis* (1990), 68–125.
[8] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*. 957–966.
[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[10] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining* (2010), 1–20.
[11] Martijn J Schuemie and Jan A Kors. 2008. Jane: suggesting journals, finding experts. *Bioinformatics* 24, 5 (2008), 727–728.
[12] Susan Trumbore, Mary-Elena Carr, and Sara Mikaloff-Fletcher. 2015. Criteria for rejection of papers without review. *Global Biogeochemical Cycles* 29, 8 (2015), 1123–1123.
[13] Shuo Yu, Jiaying Liu, Zhuo Yang, Zhen Chen, Huizhen Jiang, Amr Tolba, and Feng Xia. 2018. PAVE: Personalized Academic Venue recommendation Exploiting co-publication networks. *Journal of Network and Computer Applications* 104 (2018), 38–47.