Who Is a Better Matchmaker? Human vs. Algorithmic Judge Assignment in a High-Stakes Startup Competition

Yang Xi[1], Orelia Pi[1], Miaomiao Zhang[2], Rebecca Xiong[2], Jacqueline Lane[2], Nihar B. Shah[1],

[1]Carnegie Mellon University
[2]Harvard University
{sarinax, opi, nihars}@cs.cmu.edu, {mzhang, rxiong, jnlane}@hbs.edu

Abstract

There is growing interest in applying artificial intelligence (AI) to automate and support complex decision-making tasks. However, it remains unclear how algorithms compare to human judgment in contexts requiring semantic understanding and domain expertise. We examine this in the context of the judge assignment problem, matching submissions to suitably qualified judges. Specifically, we tackled this problem at the Harvard President's Innovation Challenge, the university's premier venture competition awarding over $500,000 to student and alumni startups. This setting represents a real-world environment where high-quality judge assignment is essential. We developed an AI-based judge assignment algorithm, the Hybrid Lexical-Semantic Similarity Ensemble (HLSE), and deployed it at the competition. We then evaluated its performance against human expert assignments using blinded match-quality scores from judges on 309 judge-venture pairs. Using a Mann-Whitney U statistic-based test, we found no statistically significant difference in assignment quality between the two approaches ($AUC = 0.48$, $p = 0.40$); on average, algorithmic matches were rated 3.90 and manual matches 3.94 on a 5-point scale, where 5 indicates an excellent match. Furthermore, manual assignments that previously required a full week could be automated in several hours by the algorithm during deployment. These results demonstrate that HLSE achieves human-expert-level matching quality while offering greater scalability and efficiency, underscoring the potential of AI-driven solutions to support and enhance human decision-making for judge assignment in high-stakes settings.

Code —
https://github.com/xasayi/Automated-Judge-Assignment

1 Introduction

The rapid growth in innovative submissions requiring evaluations, spanning academic conferences to startup competitions, has given rise to a critical challenge: how to efficiently assign expert judges to submissions when submission volume is high and the individual judge evaluation bandwidth is limited. This assignment problem requires a sophisticated understanding of both judge expertise and submission content, complicated by the need to balance workload constraints (Criscuolo et al. 2017), align domain expertise (Lane et al. 2024), and ensure evaluation quality across diverse stakeholder needs (Boudreau et al. 2016).

In entrepreneurial competitions involving evaluations of startup ventures, individuals or teams present their business ideas to expert judges with the goal of securing funding, mentoring, or other support. Commonly organized by accelerators, universities, or government agencies, these competitions often span multiple tracks or categories, where each venture receives several reviews, and each judge reviews multiple ventures based on their expertise and background.

Recent advances in automating this assignment problem have shown promising results in academic peer review settings. Computer Science (CS) conferences have been using algorithmic approaches to tackle the "reviewer assignment" problem for many years. In this setting, reviewer assignment is typically solved using a two-stage approach (Shah 2022, Section 3): (1) computing similarity scores that estimate the alignment between each potential reviewer's expertise and each submission, and (2) generating assignments based on these scores. The similarity computation process often uses publication histories or citation networks to get accurate similarity estimates (Cohan et al. 2020; Ostendorff et al. 2022). More recently, proposal reviewing agencies in astronomy have automated their reviewer assignment process (Carpenter, Corvillón, and Shah 2025). They use the history of submissions to obtain similarity estimates.

However, a significant gap exists when applying these techniques to entrepreneurial competitions, and judge assignments herein are typically performed manually by a few program administrators who possess tacit knowledge of the context and process. The best-performing algorithms for assigning reviewers to papers in peer review (OpenReview 2025), such as SPECTER2 (Singh et al. 2023) and SciNCL (Ostendorff et al. 2022), crucially rely on citation networks and past publications. However, judges in entrepreneurial competitions often lack such signals, making it much more difficult to algorithmically infer their domain knowledge. Moreover, entrepreneurial expertise spans both technical and business domains (Kacperczyk and Younkin 2017), as evaluating market potential, business viability, and implementation strategy is essential for venture startups. Existing top-performing reviewer assignment algorithms are typically built on SciBERT (Beltagy, Lo, and Cohan 2019),

a language model pre-trained on scientific corpora, which may not capture business-oriented expertise.

To address these challenges, we developed and evaluated a solution for the Harvard President's Innovation Challenge. Awarding more than $500,000 annually, this is a high-stakes real-world setting where assignment quality directly impacts outcomes. We introduce the Hybrid Lexical-Semantic Similarity Ensemble (HLSE), an ensemble similarity model that integrates three types of text representations to compute similarity estimates between judges and ventures: sparse TF-IDF vectors that contain many zeros and highlight distinctive keywords; dense transformer-based embeddings that contain mostly non-zero values and capture semantic meaning; and hybrid TF-IDF-weighted embeddings that merge both. Our approach follows the two-step pipeline used in academic peer review systems, using HLSE for similarity computation and PeerReview4All (Stelmakh, Shah, and Singh 2019) to assign matches according to the computed similarity scores. Using this system, we investigate the key question:

> *Can algorithmic judge assignments match the quality of expert human judgment in such high-stakes entrepreneurial evaluation settings?*

To answer this, we conducted the first direct empirical comparison between algorithmic and human expert assignments in entrepreneurial evaluation by collecting blinded self-reported match quality scores from judges. Overall, our contributions are threefold:

- We introduce HLSE, an ensemble that combines TF-IDF and transformer-based embeddings to calculate accurate similarity scores for the judge assignment problem.
- We find that, across 309 judge-venture matches, human expert assignments achieved a mean match quality score of 3.94 compared to 3.90 for HLSE. A statistical test based on the Mann-Whitney U statistic showed no significant difference between the two approaches based on these scores ($AUC = 0.48$, $p = 0.40$), demonstrating human-level performance in a real-world deployment.
- We demonstrate that HLSE significantly reduces assignment time from one week to several hours, offering a scalable solution to the growing demands of innovation assessment while maintaining quality standards.

2 Related Work

This work addresses the first stage of judge assignment: accurately computing similarity scores in the unique context of a large startup competition. In this section, we review existing approaches for both stages of judge assignment, examine related work evaluating similarity computation algorithms, and discuss the foundations behind our model, HLSE.

The Judge Assignment Problem

Similarity score computation. Numerous methods have been developed to find the similarity between two descriptions, primarily using Natural Language Processing (NLP) and Machine Learning (ML) techniques. Traditional approaches such as TF-IDF and topic modeling (Ferilli et al. 2006; Mimno and McCallum 2007; Anjum et al. 2019) work well on smaller datasets but often struggle to scale with large or evolving corpora.

To overcome these limitations, modern approaches increasingly utilize embedding-based methods. Sentences or documents are mapped into dense vector spaces using deep neural networks, capturing richer semantic relationships. For instance, SPECTER (Cohan et al. 2020) generates document embeddings using SciBERT (Beltagy, Lo, and Cohan 2019) fine-tuned with citation links as a weak supervision signal. These citation-informed embeddings capture semantic similarity more effectively than purely text-based methods. Other works leverage citation graph structures to further inform document similarity (Ostendorff et al. 2022). Such approaches provide richer contextual signals, particularly for documents with sparse or ambiguous text. Across all approaches, cosine similarity is commonly used for similarity computation, where a higher score indicates a more similar pair of embeddings.

Assignment algorithms. Once similarity scores have been computed, they are used to inform the assignment. A common objective is to maximize the total similarity score in all assignments to enhance global match quality. This is used in the Toronto Paper Matching System (Charlin and Zemel 2013), which has been widely adopted by major ML conferences. However, maximizing global similarity is not always sufficient for fairness and work balance. Garg et al. (2010) propose algorithms that balance judge preferences while ensuring equitable workloads and minimizing conflicts of interest. Similarly, Long et al. (2013) frame the assignment process as a topic coverage problem such that each submission is reviewed by experts covering diverse relevant topics. Building on these foundations, Kobren, Saha, and McCallum (2019) introduce a local fairness formulation that guarantees that each submission receives a minimum threshold of judge expertise. To further mitigate manipulation and fraud risks, Jecmen et al. (2020) propose algorithms that use controlled randomness in the judge assignment process, with Xu et al. (2023) later extending these algorithms.

In our work, we adopt PeerReview4All (Stelmakh, Shah, and Singh 2019), which maximizes the minimum review quality assigned to any submission. By prioritizing the most disadvantaged submissions, this approach enhances fairness globally, particularly for submissions on niche or underrepresented topics, while maintaining overall assignment quality. The PeerReview4All algorithm has previously been used for assignment of reviewers to submissions in CS conferences and astronomy proposal evaluations, resulting in strong performance (Carpenter, Corvillón, and Shah 2025; Stelmakh, Shah, and Singh 2019).

Evaluation of Similarity Computation Algorithms

Evaluating the quality of similarity computations in judge assignment remains a core challenge. In the CS peer review setting, many approaches use indirect estimates of assignment quality, such as external expertise assessment (Mimno and McCallum 2007; Zhao, Anand, and Sharma 2022). While informative, these methods can introduce noise due to

limited context or incomplete information about the judges' backgrounds. To improve accuracy, other studies collect self-reported expertise ratings. For instance, Stelmakh et al. (2025) developed a gold-standard dataset of 477 blinded paper-reviewer pairs where reviewers report their expertise on papers. This benchmark has been used by OpenReview (OpenReview 2025), a major CS peer review platform used by several flagship conferences. While valuable, this benchmark is focused on academic peer review and may not generalize to domains like entrepreneurship, where expertise signals differ. Other studies, such as the one performed by Anjum et al. (2019), gather self-reported expertise ratings from 33 reviewers after review completion to evaluate the algorithmic assignment quality.

Beyond CS, fields such as astronomy have also started adopting and evaluating algorithmic judge assignment. Recently, Carpenter, Corvillón, and Shah (2025) examined proposal review for the ALMA telescope, comparing mean similarity scores and self-reported reviewer expertise scores across multiple assignment cycles. Their findings show that automated assignment methods — specifically, topic modeling to compute similarity scores, followed by PeerReview4All (Stelmakh, Shah, and Singh 2019) — increase both similarity scores and perceived reviewer expertise while significantly reducing manual effort. This provides important empirical evidence of the effective impact of ML-based algorithmic approaches outside of CS peer review.

Given the lack of historical match-quality data in entrepreneurial settings, our experiment adopted a direct, empirical, and head-to-head comparison between algorithmic and manual assignments. Judges, blinded to the assignment source, evaluated ventures matched independently by humans or the algorithm, then reported their perceived expertise fit. This allowed us to directly assess the alignment between algorithmic and human matches across 309 judge-venture pairs. To our knowledge, this is the first such empirical comparison beyond academic peer review, offering key insights into the effectiveness of algorithmic judge assignment in a high-stakes entrepreneurial context.

Backbones of HLSE

TF-IDF methods. Introduced in 1972 (Spärck Jones 1972), TF-IDF remains a simple yet effective technique for text semantic similarity and has been widely used in a variety of applications, from text classification (Liu et al. 2018; Das and Chakraborty 2018) to information retrieval (Hiemstra 2000; Aizawa 2003; Fautsch and Savoy 2010). Term frequency (TF) measures how often a word appears in a document, while inverse document frequency (IDF) down-weights terms that appear in many documents and assigns higher weights to terms that occur in relatively few documents (Manning, Raghavan, and Schütze 2009, Section 6.2.1). This property is particularly desirable in our application, where certain highly informative words may be frequent but confined to a small subset of documents; IDF ensures such terms still receive high weight.

In judge assignment, TF-IDF has gained widespread adoption through TPMS (Charlin and Zemel 2013), which has been integrated into major ML conferences. Different venues adapt TF-IDF differently based on their needs: some rely on titles and abstracts, others use full-text. Judge profiles incorporate prior assignments and the specific TF-IDF method used can be tuned (OpenReview 2025). Despite advances in deep learning, recent studies suggest TF-IDF remains competitive (González-Márquez and Kobak 2024). Stelmakh et al. (2025) finds that TPMS with full-text inputs matches or outperforms state-of-the-art embedding models and large language models in assignment accuracy.

Transformer-based embedding methods. Embedding methods use pretrained language models to transform documents into dense vector representations. Due to the input length limitations of transformers, these methods typically rely on titles and abstracts rather than full-text (Cohan et al. 2020). Prior research shows embeddings perform better on short texts, whereas TF-IDF excels on longer texts (Meijer, Truong, and Karimi 2021). A common method used in judge assignment for CS peer review is SPECTER (Cohan et al. 2020), which uses the title and abstract of papers. It is trained with a citation-informed contrastive loss, using cited papers as positive examples and non-cited work as negative examples for semantic similarity (Cohan et al. 2020). Building on this framework, SciNCL (Ostendorff et al. 2022) uses harder examples to improve embedding quality, while SPECTER2 (Singh et al. 2023) extends the training corpus and incorporates multi-task learning. All of these models use SciBERT (Beltagy, Lo, and Cohan 2019), a variant of BERT pretrained on a large scientific corpus. However, these models remain domain-specific to scientific language and may underperform in entrepreneurial contexts, which have different vocabularies and semantic nuances.

Hybrid similarity methods. While hybrid approaches combining TF-IDF and embeddings have been explored in NLP tasks, none are used in judge assignment. Agarwal et al. (2019) and De Boom et al. (2016) compute an informed sum of TF-IDF weighted word embeddings to improve author clustering and representation learning while Didi, Walha, and Wali (2022) use a TF-IDF weighted sum of word embeddings for classification of COVID-19 tweets. Other works have explored alternative weighting schemes. Arora, Liang, and Ma (2017) use a weighted mean of word embeddings with Smoothed Inverse Frequency (SIF), defined as $a/(a + f_w)$, where a is a parameter and f_w is the frequency of word w in the corpus. While they demonstrate that SIF-based embeddings outperform neural methods such as RNNs and LSTMs on several tasks, this weighting can undervalue informative terms in our dataset that occur frequently but only within a small number of documents. All of these previous methods are based on static embeddings such as GloVe (Pennington, Socher, and Manning 2014), which cannot capture context-dependent meanings. In contrast, we use transformer-based embeddings, which model semantic nuances and can distinguish between different meanings of the same word. Consistent with this choice, Joshi et al. (2020) report that combining TF-IDF with transformer-based embeddings can yield up to a 36% relative improvement on fine-grained tasks.

Beyond using hybrid models, HLSE employs an ensemble

that integrates similarity signals from TF-IDF, transformer-based, and hybrid representations. By aggregating these signals, the ensemble reduces reliance on the strengths or weaknesses of any single representation. This strategy has been shown to often have better performance than individual models (Opitz and Maclin 1999; Dietterich 2000).

3 Problem Setup and Data

This section describes the problem setup, including the real-world deployment context of our model, the problem statement, the challenges of manual judge assignment, and an overview of the training data used for model development.

Real-world context. This work is set in the context of the Harvard President's Innovation Challenge, which annually attracts hundreds of early-stage student and alumni ventures from diverse sectors including healthcare, fintech, consumer products, and more. These ventures span various stages and compete for more than $500,000 in funding and support. All ventures undergo screening by an internal selection committee, where semifinalists are then evaluated by external judges. This external judging pool consists of high-profile judges from various institutional, functional and domain backgrounds, including experienced investors, established entrepreneurs, and senior professionals with deep technical or operational expertise. Many have led successful business exits or held senior leadership or scientist roles in top-tier firms or academic research institutions. To ensure impartiality and consistency in venture evaluation, judges are assessed for relevance to ventures and provided with structured rubric criteria to standardize evaluations.

Problem statement. For the 2025 competition, 231 judges and 101 semifinalist ventures were considered for automated assignment. The core task is to assign each venture to a panel of qualified judges whose expertise, background, and experience align with the venture's industry, technology, and market focus. At the same time, the assignment must satisfy logistical and fairness constraints, including:

- Load balancing: Each judge can evaluate at most 7 ventures and each venture must have 12 judge assignments.
- Track constraints: Judges are matched only to ventures in the same tracks ("Open", "Social Impact", "Healthcare and Life Sciences").
- Conflict of interest exclusions: Assignments where a judge has personal, professional, or financial ties to a venture must be avoided.

These constraints align with competition goals and ensure standardization compared to previous practice (see Table 1).

Limitations of manual process. From 2021 to 2024, the assignment of judges to ventures was performed manually by an internal program administrator using informal heuristics. While this approach benefited from continual assignment experience and institutional knowledge, it became increasingly unsustainable due to several reasons:

- *Growing scale*: The number of ventures grew from 112 to 133 and judges from 201 to 281. This expanded the potential matching space by 66%, from 22,512 (112×201) to 37,373 (133×281).
- *Staff turnover*: Knowledge of past judge-venture matching decisions was difficult to retain and transfer between program administrators.
- *Heuristic-based groupings*: manual groupings used by the program administrator prioritized perceived quality within small clusters rather than global semantic alignment. Prior work has shown that grouping-based constraints, whether for diversity (Benabbou et al. 2020), two-phase review design (Jecmen et al. 2022), or strategy proofing in peer review (Dhull et al. 2022), can reduce overall match quality compared to individualized matching.

Year	Ventures per Judge			Judges per Venture		
	Min	Mean	Max	Min	Mean	Max
2021	2	7	12	8	13	22
2022	2	7	12	8	12	17
2023	1	8	10	7	16	27
2024	2	8	13	9	16	23

Table 1: Historical venture-judge assignments summary statistics.

Manually constructing such an assignment on this scale risks inferior match quality and fairness concerns. These challenges, combined with the need for scalable and high-quality judge-venture matches, highlight the need for a computationally-driven solution.

Training data. To train our assignment model, we received four years (2021-2024) of historical competition data from the organizers, consisting of three components:

- Venture applications, encompassing pitch descriptions, track, venture industry, venture problem description, and solution details.
- Judge profiles, containing short bios, preferred track, areas of expertise, and industry interests. If the profile length contained fewer than 50 words, we supplemented it with publicly available data collected and verified by the competition organizers, sourced from their LinkedIn profiles and company websites.
- Historical assignments, consisting of 1,400 to 2,100 judge-venture matches each year.

While these records could serve as training data, using venture and judge text descriptions as inputs and past assignments as labels, the informal manual heuristics underlying these past matches make them unreliable as ground truth. Moreover, the historical assignments provide only a binary signal indicating whether a judge-venture pair was assigned, with no measure of match quality. To create a more reliable and informative training dataset, we manually scored a subset of 105 judge-venture pairs for match quality on a 1-5 scale, where 1 denotes no relevant judge expertise alignment, and 5 reflects precise alignment of judge expertise with the venture's area. These scores were assigned without relying on any pre-existing groupings by members of

the research team as well as the competition staff, both of whom have deep knowledge of the competition context and assignment process. This curated dataset served as a trusted ground-truth dataset for match quality to develop our model.

In addition, to prepare the input judge profile and venture application text description for similarity computations, we only retained a subset of relevant fields from the raw competition data. Then, we sanitized the input text using established methods (Charlin and Zemel 2013; Xu et al. 2019).

4 HLSE Method

The HLSE method is an ensemble model that integrates multiple similarity computation models as base learners. These base learners include TF-IDF models, transformer-based embedding models, and hybrid models that combine IDF weighting with transformer-based representations. In this section, we first describe two hybrid models used in HLSE: one based on document-level similarity and the other on token-level similarity. We then outline the specific base models used. Finally, we explain how the outputs of all base learners are combined into a unified similarity ensemble.

Document level mean hybrid similarity. Let $T \in \mathbb{R}^{d\times L}$ denote the transformer-based token embeddings with a sequence length L and embedding dimension d. Further, let $w \in \mathbb{R}^{L}_{>0}$ denote IDF weights corresponding to each token. Using this notation, we compute the document embedding, D, for each input (venture application or judge profile) as: $D = \frac{T\cdot w}{\|w\|_1}$. Given such embeddings for any two documents $D^{(1)}$ and $D^{(2)}$, we use cosine similarity to evaluate the semantic similarity between them: $\frac{D^{(1)}\cdot D^{(2)}}{\|D^{(1)}\|_2\cdot\|D^{(2)}\|_2} \in \mathbb{R}$. This yields a value in $[-1, 1]$, where 1 denotes maximum similarity and -1 denotes maximum divergence.

Token level mean hybrid similarity. While the document-level method aggregates embeddings before similarity calculation, the token-level approach calculates similarity at the token stage and then applies the IDF. This better compares how closely individual tokens from different documents align. For two token embedding columns $T^{(1)}_i$ at the i^{th} column in the first document and $T^{(2)}_j$ at the j^{th} column in the second document, we calculate the IDF-weighted cosine similarity between them $s_{i,j} = \frac{T^{(1)}_i\cdot T^{(2)}_j}{\|T^{(1)}_i\|_2\cdot\|T^{(2)}_j\|_2}\cdot w^{(1)}_i w^{(2)}_j \in \mathbb{R}$. We then compute the overall similarity over the set $\mathcal{S}$ of all token level similarities: $\frac{1}{|\mathcal{S}|}\sum_{s_{i,j}\in\mathcal{S}} s_{i,j} \in \mathbb{R}$. This yields a similarity score where a larger positive value indicates that the two documents are more similar.

Specific base learners. HLSE consists of three types of base learners: TF-IDF models, transformer-based embedding models, and hybrid models. For TF-IDF, we implement two approaches, each trained with and without an expanded Wikipedia corpus to enhance IDF generalization, resulting in 4 variants. The first approach is standard TF-IDF with IDF smoothing. The second is an augmented TF-IDF approach by (Xu et al. 2019), which mitigates the bias toward longer documents through an adjusted TF and does not use IDF smoothing. The supplementary Wikipedia corpus, retrieved via WikipediaAPI, contains 2,108 documents with a mean length of 250 words and standard deviation of 29 words.

Base Type	Specific method
TF-IDF	TFIDF with Wikipedia Weighted IDF
Embedding	BERT Token Mean Pooled
Embedding	MPNet Document Mean Pooled
Embedding	RoBERTa Document Mean Pooled
Hybrid	BERT IDF Weighted Token Pool

Table 2: The base models in the ensemble learner.

For embedding models, we use 4 transformer models. These include a general-purpose model, BERT, and more specialized sentence transformers like RoBERTa and MPNet. We also selected the top-performing model on the MTEB (Muennighoff et al. 2023) Semantic Textual Similarity (STS) benchmark, prioritizing resource-efficient architectures due to limited local compute. Notably, *GIST-Embedding-v0* is a lightweight model with a sub-1 GB memory footprint and no multilingual tuning, with strong performance on STS tasks. For each of the embedding models, we derive 2 base learners by computing similarity at either the document level or the token level. For hybrid models, we combine the two hybrid similarity computation schemes detailed earlier with four IDF schemes, yielding 8 variants for each embedding model. In addition to these models, we also included GPT-4o using zero, one, and few shot prompting as detailed in Appendix A. In total, we have 4 TF-IDF base learners, 8 embedding base learners, 32 hybrid base learners, and 3 GPT-4o base learners.

Ensemble learning. After computing similarity scores from the base models, we combine them into a single ensemble similarity score. Our goal is to learn nonnegative weights such that the weighted combination of base similarity scores best approximates the ground truth match quality scores. We achieve this by solving a linear regression problem with convexity constraints, requiring the nonnegative weights to sum to one. The ensemble weights are learned by minimizing the mean squared error between the ground-truth labels and the weighted predictions. We perform five-fold cross-validation, and the final ensemble prediction is obtained as a weighted sum of the base model similarity scores using the learned weights. Only 5 of 47 base learners received weights greater than 0.01 in every fold. These five models were retained in the final ensemble, with the remaining dropped. The specific models that make up the ensemble are shown in Table 2. On the match quality training dataset, 85% of the ensemble's predicted scores fall within one point of the ground-truth, and 37% match the ground truth exactly.

5 Algorithmic Versus Manual Assignment Experiment

To compare the quality of algorithmic and manual judge assignments, we conducted an experiment based on the

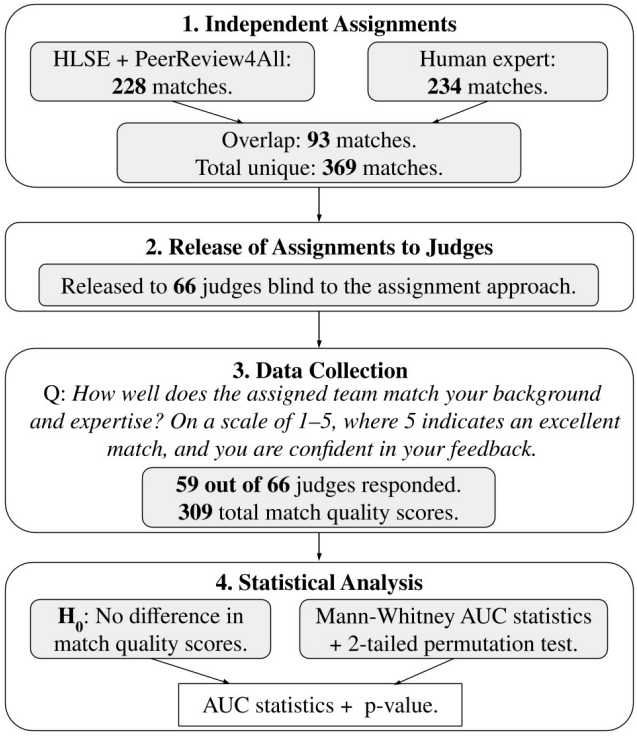

Figure 1: Algorithmic versus manual experiment.

judges' self-reported assessment of expertise alignment with the ventures they were matched with. Figure 1 outlines the experimental workflow.

Experiment design. To isolate the effect of the assignment method, manual and algorithmic assignments were performed independently. Per the competition organizers' request, the manual assignment was only performed in one of the tracks. This track was the smallest in size, consisting of 19 ventures and 66 judges. Each venture in this experiment track received 11 to 13 manual judge assignments and exactly 12 algorithmic assignments. This resulted in 369 unique judge-venture pairs, with 228 algorithmic matches, 234 manual matches, and 93 overlapping matches. All such algorithmic matches were accepted by the organizers without adjustment. In addition, judges were blind to the assignment method. For the other two tracks, which used algorithmic assignment but were not part of this experiment, the competition organizers performed manual refinement on the algorithmic assignments before releasing them to the judges.

Outcome measurement. Judges were asked the following question with radio buttons ranging from 1 to 5:

> *How well does the assigned team match your background and expertise? On a scale of 1-5, where 5 indicates an excellent match, and you are confident in your feedback.*

To ensure consistency with other questions asked in the evaluation form unrelated to this experiment, the order of the options is not randomized. Of the 66 judges who indicated a preference to participate in the experiment track, 59 provided responses, yielding 309 total match quality scores.

Statistical test. To compare the two methods, we tested the null hypothesis that there is no systematic difference in match quality between algorithmic and manual assignments. We use a test based on the Mann-Whitney U statistic, a nonparametric method for comparing ordinal data from independent groups. Our setup considers each venture as a unit of comparison, with judges either assigned manually or algorithmically. To find statistical significance, we performed a two-tailed permutation test with details in Appendix B.

6 Results

We deployed HLSE to compute the similarity scores between judges and ventures in the 2025 competition. In this section, we analyze results from the experimental procedure, the time taken for manual versus algorithmic assignment, and an internal post-deployment audit.

Experiment results. Following the experimental procedure in the previous section, we collected self-reported match quality scores for 309 judge-venture pairs. Table 3 summarizes the comparison between manual and algorithmic assignments. The mean self-reported match quality scores are comparable: 3.94 for manually assigned matches and 3.90 for algorithmically assigned matches. The statistical test yields an AUC of 0.48 and a p-value of 0.40. Here, an AUC of 0.5 indicates no difference in score distribution between the two groups, while an AUC closer to 0 or 1 would indicate that one group always has higher scores than the other. These results indicate no statistically significant differences in match quality between the assignment approaches. In addition, we observe that there are 82 overlapping matches in which judge-venture pairs were assigned by both HLSE and the human expert. These matches have a mean score of 4.10.

Manual versus algorithmic assignment time. We compared the time required for algorithmic versus manual assignments. Manual times were provided by the competition organizers, while algorithmic times included both the actual runtime of the algorithm (on an Apple M1 chip with 16GB of RAM) and the time it took to scrape publicly available supplementary data for judges. This data collection process was performed using automated tools by the competition organizers and is necessary each year because the information can change over time. The time required to write the scrapers and develop the algorithm is not included since it is an amortized cost, and the developed code can be reused.

	Manual	Algorithmic
Number of scores	197	194
Mean score	3.94 ($\pm$0.93)	3.90 ($\pm$0.94)
Number of overlaps	82 (included in both)	
Mean score for overlaps	4.10 ($\pm$0.94)	

Table 3: Summary mean and standard deviation of the match quality scores from manual versus algorithmic assignments.

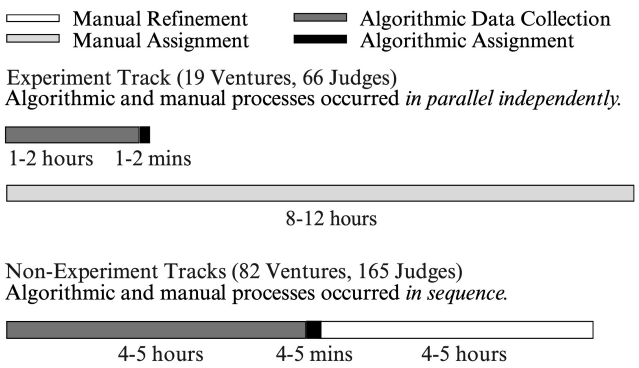

Figure 2: Overview for time taken for 2025 assignments.

For 2021 to 2024, the manual process took about *a full week* each year. This included reading judge and venture descriptions, searching online for additional data about judges' expertise, and manually matching judges to ventures. Figure 2 summarizes the time spent on assignments for 101 ventures and 231 judges in 2025, split between the experiment and non-experiment tracks. In 2025, manual and algorithmic assignments were performed independently on the experiment track. Manual assignment took 8-12 hours, while algorithmic data collection and assignment took 1-2 hours and 1-2 minutes, respectively. For the non-experiment tracks, algorithmic data collection took 4-5 hours and algorithmic assignment took 4-5 minutes, followed by an additional 4-5 hours of manual refinement.

Although the experiment track was much smaller in scale, its manual assignment alone took roughly the same amount of time as the combined manual refinement and algorithmic assignment across non-experiment tracks. Moreover, our experiment results show that algorithmic and manual approaches achieved comparable match quality, indicating that HLSE can generate reliable similarity scores that maintain assignment quality without refinement. Compared to the week-long process in previous years, the algorithmic approach reduced the total time to approximately 5-7 hours.

Internal audit. Post-deployment, we conducted an internal audit in which we discovered a minor implementation flaw affecting all hybrid models due to a tokenization mismatch. While this flaw introduced some noise in the models' similarity estimates, it did not meaningfully change the ensemble's overall behavior. We recalculated the AUC statistic, obtaining 0.49 with a p-value of 0.76, indicating no statistically significant difference between algorithmic and manual assignments. Comparing the old against the new algorithmic matches, we find that there are 168 overlapping matches, with a mean match quality score of 3.92 ($\pm$0.95). The 26 matches assigned only by the old ensemble had a mean score of 3.77 ($\pm$0.80), while the 5 matches assigned only by the new ensemble had a mean score of 4.4 ($\pm$0.80). While there is some difference in the assigned judge-venture pairs, removing the flawed hybrid model did not produce a statistically significant change in the distribution of match quality scores, and the mean match quality scores remain comparable. Unintentionally, these results demonstrate that our algorithm is robust to noise in the base learners.

7 Conclusions

Assigning appropriate judges to submissions is a critical challenge in various domains that require high-quality evaluation. Existing similarity algorithms deployed for reviewer assignment in peer review are not directly applicable to other domains. In this work, we tackle judge assignment in the context of a competitive entrepreneurial challenge. We developed an ensemble similarity model, HLSE, and deployed it in the Harvard President's Innovation Challenge, a venture competition that awards $500,000 in prizes annually. Our results demonstrate that algorithmic judge assignment performs on par with expert humans in this entrepreneurial context. Matches produced by HLSE are statistically indistinguishable from matches produced by human experts based on self-reported match quality scores. Furthermore, the algorithmic approach is significantly more scalable and time efficient, reducing assignment time from a full week to hours. Below, we discuss key lessons from HLSE's deployment at the 2025 competition and outline directions for future work.

Curating ground truth datasets. Early in development, we observed that historical assignment data were often inconsistent or suboptimal. This reflects a common challenge in real-world settings, where the available data can be noisy and incomplete. Such issues can significantly hinder meaningful evaluation. To address this, we curated a high-quality, expert-informed match quality dataset using insight and expertise from the organizers, providing a more reliable and consistent ground truth dataset to develop our method. This experience highlights the importance of developing reliable ground truth datasets that reflect real-world expectations, especially when existing data is unreliable or unavailable.

Empirical stability of the ensemble. A post-deployment audit uncovered a minor implementation flaw in the tokenization scheme for hybrid models, introducing some noise to the similarity estimates. Despite this, the ensemble's overall performance remained stable, and most individual matches remained consistent between the old and revised ensembles. These results indicate that, in this deployment, the ensemble performance was largely unaffected by the flaws in a subset of base models, (unintentionally) providing empirical evidence of system stability in practice.

Human-AI collaboration insight. Our experiment showed that algorithmic assignments using HLSE for computing similarity scores and PeerReview4All (Stelmakh, Shah, and Singh 2019) for assignment performed on par with manual expert assignments. Importantly, even with human refinement for the non-experiment tracks, the combined assignment process remained significantly faster than full manual assignment. This suggests that AI-assisted workflows can effectively balance scalability and efficiency with human oversight, offering a practical path to high-quality assignments.

Future work. Building on the lessons learned, we propose several directions to address current limitations.

- *Expanded cross-domain and longitudinal validation:* To evaluate the generalization of HLSE, future work can extend the algorithmic versus human evaluation across a larger sample size, more competition tracks, and different types of review systems (e.g., hackathons). Deploying HLSE and PeerReview4All (Stelmakh, Shah, and Singh 2019) over multiple annual cycles and varying institutional contexts would allow a better assessment of the stability of the method under different data regimes, judge pools, and domain characteristics. Finally, this can help isolate HLSE's performance from contextual effects tied to any single deployment.
- *Prevent gaming of automated assignment methods:* Recent work (Eisenhofer et al. 2023; Hsieh, Raghunathan, and Shah 2025) has shown that automated assignment algorithms are susceptible to manipulation, whereby strategic text modifications can be used to exploit the method and obtain favorable matches. Such vulnerabilities can facilitate fraudulent activities, including collusion rings (Littman 2021; Jecmen et al. 2020) and identity theft (Shah et al. 2025). This poses a serious risk to the integrity of such algorithms, especially in high-stakes settings such as venture competitions. Future work can investigate how to protect HLSE against such adversarial behavior, building on the ideas from (Hsieh, Raghunathan, and Shah 2025; Jecmen et al. 2020; Shah et al. 2025) that are deployed in CS conference reviewing.
- *Human-in-the-loop feedback:* Building on current results, we could investigate how integrating human input into algorithmic assignments would affect match quality and efficiency. For instance, one could explore human-refined algorithmic assignments compared with fully manual or algorithmic assignments to better understand the specific benefits of human-AI collaboration.

A Using GPT-4o as a Judge

We queried *gpt-4o-2024-08-06* with a temperature of 0 (no randomness) to produce similarity scores given a venture and a judge. We used zero, one, and few shot learning with the following prompt:

```
1 You are a manual assigner who is
      responsible for assigning judges to
      ventures for a venture competition.
      You are given two blocks of text, D1
      being the venture and D2 being the
      judge. You need to assess whether the
       judge is a good match to the venture
       considering whether the judge has
      technical expertise in the venture's
      area. Rate the match on a scale of
      1-5 (5 is good, 1 is bad).
      Internalize your reasoning and only
      output a number between 1 to 5.
2 D1: <description of the venture>
3 D2: <description of the judge>
```

We compare the scores generated by GPT-4o against the ground truth and find the Kendall Tau-b rank correlation, τ, and the associated p-value, p, between them. The null hypothesis for the p-value is that the generated and ground-truth scores are independent. For zero-shot, we obtain $\tau = -0.02$ ($p = 0.822$); for one-shot, $\tau = 0.00$ ($p = 0.980$); and for few-shot (two examples), $\tau = 0.02$ ($p = 0.752$).

B Statistical Test

We provide details about the statistical test for the algorithmic versus manual experiment.

Mann-Whitney U statistic. We compare match quality between algorithmic and manual assignments using an adjusted Mann-Whitney U statistic to avoid double counting the overlapping algorithmic and manual assignments. For each venture $i \in \{1, \dots, k\}$, let

- A_i: set of algorithm-only scores, $n_i^A = |A_i|$,
- M_i: set of manual-only scores, $n_i^M = |M_i|$,
- O_i: set of overlapping scores, $n_i^O = |O_i|$.

where the scores are in $\{1, \dots, 5\}$. The Mann-Whitney score function between two scores x, y is $h(x, y) = \mathbb{I}[x > y] + \frac{1}{2}\mathbb{I}[x = y]$, where $\mathbb{I}[\cdot]$ is the indicator function. Then, the per-venture U statistic is

$$U_i = \sum_{a \in A_i \cup O_i} \sum_{m \in M_i \cup O_i} h(a, m).$$

Since the overlapping assignments contribute equally to both groups and can artificially inflate the U statistics, we subtract the self-comparison term:

$$U_i^o = \sum_{o_1 \in O_i} \sum_{o_2 \in O_i} h(o_1, o_2).$$

The overlap-adjusted per venture AUC statistic is:

$$T_i = \frac{U_i - U_i^o}{V_i}, \quad V_i = (n_i^A + n_i^O)(n_i^M + n_i^O) - n_i^O \cdot n_i^O,$$

where V_i is the maximum possible adjusted U statistic for venture i. When $n_i^O = 0$, this reduces to the standard AUC: $T_i = U_i/(n_i^A \cdot n_i^M)$. Finally, we combine ventures via an inverse-variance weighted mean: $T = \frac{\sum_{i=1}^k T_i \cdot 1/\sigma_i^2}{\sum_{i=1}^k 1/\sigma_i^2}$ where σ_i^2 is the empirical variances of all scores for venture i. This weighting ensures that ventures with more stable estimates contribute more to the overall statistic.

Two-tailed permutation test. To evaluate the statistical significance of the weighted AUC statistic T, we employ a two-tailed permutation test with $N = 5000$ resamples. For each venture i, we keep overlapping assignments fixed and randomly permute the labels of the other assignments. This simulates the null hypothesis that the assignment method has no effect on match quality. For each permutation $j \in \{1, \dots, N\}$, we compute T^j. The two-sided p-value, p, is:

$$p = \frac{1 + \sum_{j=1}^N \mathbb{I}\big[\,|T^j - 0.5| \geq |T - 0.5|\big]}{1 + N},$$

which represents the proportion of permutations in which the deviation from 0.5 is at least as extreme as the observed deviation. An AUC value of 0.5 indicates that the assignment method is equally likely to be manual or algorithmic.

Acknowledgments

This work was done under the approval of the Carnegie Mellon University Institutional Review Board. We gratefully acknowledge the contributions and insights of the organizing team and collaborators at Harvard Innovation Labs. This work was supported in part by NSF 1942124 and ONR N000142512346.

References

Agarwal, L.; Thakral, K.; Bhatt, G.; and Mittal, A. 2019. Authorship Clustering using TF-IDF weighted Word-Embeddings. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, 24–29. New York, NY, USA: Association for Computing Machinery. ISBN 9781450377508.

Aizawa, A. 2003. An information-theoretic perspective of TF-IDF measures. *Information Processing & Management*, 39(1): 45–65.

Anjum, O.; Gong, H.; Bhat, S.; Hwu, W.-M.; and Xiong, J. 2019. PaRe: A Paper-Reviewer Matching Approach Using a Common Topic Space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 518–528. Hong Kong, China: Association for Computational Linguistics.

Arora, S.; Liang, Y.; and Ma, T. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations*.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics.

Benabbou, N.; Chakraborty, M.; Ho, X.-V.; Sliwinski, J.; and Zick, Y. 2020. The Price of Quota-based Diversity in Assignment Problems. *ACM Transactions on Economics and Computation*, 8(3): 1–32.

Boudreau, K. J.; Guinan, E. C.; Lakhani, K. R.; and Riedl, C. 2016. Looking Across and Looking Beyond the Knowledge Frontier: Intellectual Distance, Novelty, and Resource Allocation in Science. *Manage. Sci.*, 62(10): 2765–2783.

Carpenter, J. M.; Corvillón, A.; and Shah, N. B. 2025. Enhancing Peer Review in Astronomy: A Machine Learning and Optimization Approach to Reviewer Assignments for ALMA. *Publications of the Astronomical Society of the Pacific*, 137(3): 034501.

Charlin, L.; and Zemel, R. S. 2013. The Toronto Paper Matching System: An automated paper-reviewer assignment system. In *ICML*.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2270–2282. Online: Association for Computational Linguistics.

Criscuolo, P.; Dahlander, L.; Grohsjean, T.; and Salter, A. 2017. Evaluating Novelty: The Role of Panels in the Selection of R&D Projects. *Academy of Management Journal*, 60(2): 433–460. Posted on SSRN July 3, 2017. Available at SSRN abstract 3650896.

Das, B.; and Chakraborty, S. 2018. An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. arXiv:1806.06407.

De Boom, C.; Van Canneyt, S.; Demeester, T.; and Dhoedt, B. 2016. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, 80: 150–156.

Dhull, K.; Jecmen, S.; Kothari, P.; and Shah, N. B. 2022. Strategyproofing Peer Assessment via Partitioning: The Price in Terms of Evaluators' Expertise. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 10(1): 53–63.

Didi, Y.; Walha, A.; and Wali, A. 2022. COVID-19 tweets classification based on a hybrid word embedding method. *Big Data and Cognitive Computing*, 6(2): 58.

Dietterich, T. G. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, 1–15. Berlin, Heidelberg: Springer-Verlag. ISBN 3540677046.

Eisenhofer, T.; Quiring, E.; Möller, J.; Riepel, D.; Holz, T.; and Rieck, K. 2023. No more reviewer# 2: Subverting automatic {Paper-Reviewer} assignment using adversarial learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, 5109–5126.

Fautsch, C.; and Savoy, J. 2010. Adapting the TF-IDF vector-space model to domain specific information retrieval. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, 1708–1712. New York, NY, USA: Association for Computing Machinery. ISBN 9781605586397.

Ferilli, S.; Di Mauro, N.; Basile, T. M. A.; Esposito, F.; and Biba, M. 2006. Automatic topics identification for reviewer assignment. In *Proceedings of the 19th International Conference on Advances in Applied Artificial Intelligence: Industrial, Engineering and Other Applications of Applied Intelligent Systems*, IEA/AIE'06, 721–730. Berlin, Heidelberg: Springer-Verlag. ISBN 3540354530.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2010. Assigning Papers to Referees. *Algorithmica*, 58(1): 119–136.

González-Márquez, R.; and Kobak, D. 2024. Learning representations of learning representations. In *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR): Harnessing Momentum for Science*.

Hiemstra, D. 2000. A probabilistic justification for using TF×IDF term weighting in information retrieval. *International Journal on Digital Libraries*, 3: 131–139.

Hsieh, J.-Y.; Raghunathan, A.; and Shah, N. B. 2025. Vulnerability of Text-Matching in ML/AI Conference Reviewer Assignments to Collusions. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*.

Jecmen, S.; Zhang, H.; Liu, R.; Fang, F.; Conitzer, V.; and Shah, N. B. 2022. Near-Optimal Reviewer Splitting in Two-Phase Paper Reviewing and Conference Experiment Design. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, 1642–1644. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

Jecmen, S.; Zhang, H.; Liu, R.; Shah, N. B.; Conitzer, V.; and Fang, F. 2020. Mitigating manipulation in peer review via randomized reviewer assignments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. ISBN 9781713829546.

Joshi, B.; Shah, N.; Barbieri, F.; and Neves, L. 2020. The Devil is in the Details: Evaluating Limitations of Transformer-based Methods for Granular Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3652–3659. International Committee on Computational Linguistics.

Kacperczyk, A.; and Younkin, P. 2017. The Paradox of Breadth: The Tension between Experience and Legitimacy in the Transition to Entrepreneurship. *Administrative Science Quarterly*, 62(4): 731–764.

Kobren, A.; Saha, B.; and McCallum, A. 2019. Paper Matching with Local Fairness Constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 1247–1257. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.

Lane, J. N.; Boussioux, L.; Ayoubi, C.; Chen, Y. H.; Lin, C.; Spens, R.; Wagh, P.; and Wang, P.-H. 2024. Narrative AI and the Human-AI Oversight Paradox in Evaluating Early-Stage Innovations. Technical Report 25-001, Harvard Business School, Technology & Operations Management Unit, Boston, MA. Revised May 2025.

Littman, M. L. 2021. Collusion rings threaten the integrity of computer science research. *Communications of the ACM*, 64(6): 43–44.

Liu, C.-z.; Sheng, Y.-x.; Wei, Z.-q.; and Yang, Y.-Q. 2018. Research of Text Classification Based on Improved TF-IDF Algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, 218–222.

Long, C.; Wong, R. C.-W.; Peng, Y.; and Ye, L. 2013. On Good and Fair Paper-Reviewer Assignment. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM)*, 1145–1150. IEEE.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2009. *An Introduction to Information Retrieval*. Cambridge University Press. Online PDF.

Meijer, H. J.; Truong, J.; and Karimi, R. 2021. Document Embedding for Scientific Articles: Efficacy of Word Embeddings vs TF-IDF. arXiv:2107.05151.

Mimno, D.; and McCallum, A. 2007. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, 500–509. New York, NY, USA: Association for Computing Machinery. ISBN 9781595936097.

Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2023. MTEB: Massive Text Embedding Benchmark. arXiv:2210.07316.

OpenReview. 2025. Expertise Modeling for the OpenReview Matching System. https://github.com/openreview/openreview-expertise. Accessed: 2025-06-05.

Opitz, D.; and Maclin, R. 1999. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11: 169–198.

Ostendorff, M.; Rethmeier, N.; Augenstein, I.; Gipp, B.; and Rehm, G. 2022. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11670–11688. Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.

Shah, N. B. 2022. Challenges, experiments, and computational solutions in peer review. *Commun. ACM*, 65(6): 76–87.

Shah, N. B.; Bok, M.; Liu, X.; and McCallum, A. 2025. Identity Theft in AI Conference Peer Review. *Communications of the ACM*. ArXiv:2508.04024.

Singh, A.; D'Arcy, M.; Cohan, A.; Downey, D.; and Feldman, S. 2023. SciRepEval: A Multi-Format Benchmark for Scientific Document Representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5548–5566. Singapore: Association for Computational Linguistics.

Spärck Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1): 11–21.

Stelmakh, I.; Shah, N. B.; and Singh, A. 2019. PeerReview4All: Fair and Accurate Reviewer Assignment in Peer Review. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, 828–856. PMLR.

Stelmakh, I.; Wieting, J. F.; Xi, Y.; Neubig, G.; and Shah, N. B. 2025. A Gold Standard Dataset for the Reviewer Assignment Problem. *Transactions on Machine Learning Research*.

Xu, Y.; Zhao, H.; Shi, X.; and Shah, N. B. 2019. On strategyproof conference peer review. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, 616–622. AAAI Press. ISBN 9780999241141.

Xu, Y. E.; Jecmen, S.; Song, Z.; and Fang, F. 2023. A one-size-fits-all approach to improving randomness in paper assignment. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Zhao, Y.; Anand, A.; and Sharma, G. 2022. Reviewer Recommendations Using Document Vector Embeddings and a Publisher Database: Implementation and Evaluation. *IEEE Access*, 10: 21798–21811.