Exploring the Influence of Structural Features and Metadata in Predicting Conference Rankings for Academic Papers

FRANCOIS Jérôme
Supervisor

KAMIYA SVIERK Vinicius
Student

M'HAMDI Amine
Student

PLONTZ Nathan
Student

***Abstract*—The exponential growth of scientific literature has made the evaluation and ranking of academic papers increasingly challenging for peer reviewers. This paper addresses the problem of automatically predicting appropriate conference rankings (A*, A, B, C, or unranked) for scientific papers based on structural features and metadata without performing in-depth content analysis. We develop a comprehensive pipeline that combines arXiv metadata extraction, PDF structural analysis, and supervised machine learning techniques to classify papers according to their potential publication venue quality. Using a dataset of 6,282 computer science papers enriched with CORE conference rankings, we train and evaluate several classification models, with Random Forest achieving the best performance. Despite our extensive feature engineering efforts, our experimental results demonstrate that structural features alone—such as reference count, abstract length, and readability scores—have limited capacity to reliably distinguish between different conference tiers, highlighting the complexity of quality assessment in scientific literature. This investigation offers valuable insights into the challenges of automated quality evaluation while raising important questions about the relationship between paper structure and research impact.**

I. Introduction

Over the past few decades, the scientific community has witnessed an unprecedented surge in the number of research articles produced worldwide. This remarkable growth stems from various factors: the globalization of academic research, increasing pressure on researchers to publish, and the rise of digital platforms facilitating knowledge dissemination.

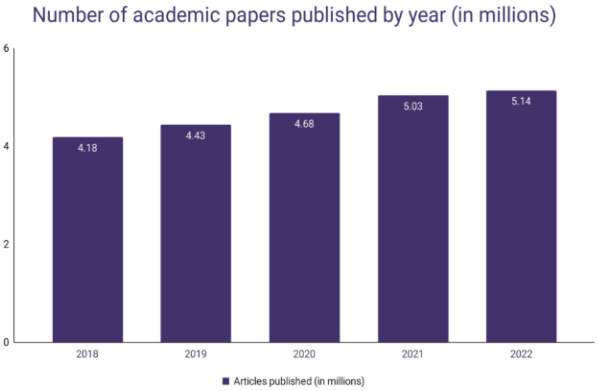

Fig. 1. Number of academic papers published by year [1]

Historically, scientific publications underwent a rigorous peer-review process before reaching the public. This procedure, established as the standard for ensuring the validity and quality of academic research, involves expert reviewers evaluating the methodology, accuracy, and significance of submitted work. While essential for maintaining scientific integrity, the peer-review process is often criticized for being slow, complex, and sometimes opaque.

To address this issue, preprint platforms emerged as a fast and open alternative. Among them, arXiv, launched in 1991 for the physics community, has grown into one of the most prominent repositories of scientific preprints, now covering fields such as mathematics, computer science, and quantitative biology. ArXiv embodies the principles of open access, enabling researchers to share their work freely and rapidly, without waiting for formal publication. Today, the platform hosts over two million articles, with thousands of new submissions added each month. This sheer volume has made arXiv an invaluable resource but also an overwhelming one for researchers trying to keep up with the latest developments in their field.

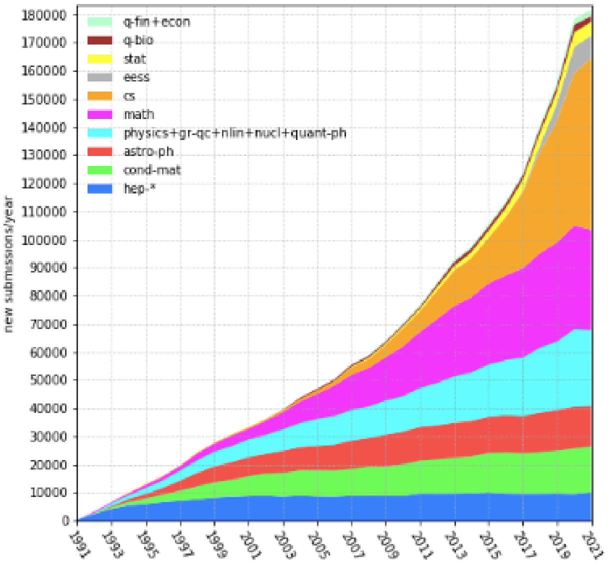

Fig. 2. arXiv submission rate statistics [2]

The increasing mass of preprints raises a critical challenge: how can researchers assess the quality and relevance of so many articles efficiently? While many preprints represent rigorous and impactful work, others may be preliminary or lack scientific soundness, aiming mainly to establish priority. Manual review at scale is unfeasible, prompting interest in Artificial Intelligence (AI) tools to assist with the evaluation of scientific texts. In particular, Large Language Models (LLMs) such as GPT or LLaMA have recently shown great potential in understanding and analyzing complex natural language, including academic writing. These AI-driven models could streamline several stages of the evaluation process: identifying methodological flaws, summarizing key findings, detecting potential ethical issues, and even estimating the impact of research. Such applications could accelerate the peer-review process.

Although prior studies have explored using AI to assess article quality or assist in peer review, the results have been mixed and remain relatively underexplored. While AI has shown promise in automating certain tasks like detecting patterns or extracting key information, challenges persist in designing systems capable of accurately assessing the depth and significance of complex academic work. The integration of AI into academic publishing remains in its early stages, and further research is needed to unlock its full potential in enhancing the peer-review process.

II. Related Work

Evaluating the quality of scientific articles has long been a central concern for researchers, editors, and institutions. While traditional peer review remains a cornerstone of scholarly validation, its subjectivity and inconsistency have prompted the development of complementary, automated methods. In recent years, a growing body of research has sought to quantify and enhance scientific quality assessment using computational approaches. The following works highlight key directions in this evolving landscape, ranging from bibliometric analyses to argument mining and the use of large language models.

The study by Szklo (2006) [3] lays the groundwork by highlighting key aspects of scientific article quality. His work focuses on the main evaluation criteria: scientific contribution, clarity, methodological robustness, and the structure of articles. He identifies common pitfalls in scientific writing across sections like the abstract, introduction, methods, results, and discussion. For instance, Szklo emphasizes the need for clear and concise abstracts, well-justified study rationales, and detailed methodological descriptions. Furthermore, he underlines the importance of post-publication evaluation, stressing that citation counts are not always indicative of quality due to biases such as self-citations or popularity over scientific merit.

Building on this foundation, Yu et al. (2014) [4] introduce a stepwise regression analysis to predict the citation impact of scientific papers. Their model incorporates bibliometric features such as the number of references, the number of authors, and the impact factor of the journal. While the study primarily focuses on citation prediction, it also uncovers useful indicators of article visibility and relevance that can be leveraged for quality assessment. However, as highlighted in the presentation, the focus on citation as a sole indicator is limited, since early visibility does not always correlate with long-term academic impact.

Cheng et al. (2020) [5] address this limitation by introducing APE (Argument Pair Extraction), a multi-task learning framework designed to extract argumentative structures from peer reviews and rebuttals. This model allows for the identification of argumentative strengths and weaknesses in review dialogues, providing a deeper understanding of how reviewers critique scientific contributions. Complementing this approach, Singh et al. (2021) [6] develop COMPARE, a taxonomy and dataset focused on comparison discussions in peer reviews. Their work emphasizes how comparative analysis in reviews can highlight novelty and significance, elements often missed by purely citation-based metrics.

Further advancing the analysis of peer review content, Hua et al. (2019) [7] apply argument mining techniques to dissect the structure and content of peer reviews. They identify various types of argumentative propositions, such as evaluations, requests, facts, references, and quotes, within a large corpus of reviews. Their findings reveal that understanding the distribution and function of these propositions can provide insights into the effectiveness and focus of reviews, highlighting the variability of argumentative structures across different venues and topics.

The potential of large language models (LLMs) for automated review generation is explored by Wu et al. (2024) [8], who propose an LLM-based framework for drafting peer reviews. Despite the promising fluency of generated reviews, Zhou et al. (2024) [9] and Liu Shah (2023) [10] reveal substantial limitations. These models tend to produce overly positive and generic feedback, lacking the nuanced critique that characterizes expert reviews. This suggests that while LLMs can support review generation, they are not yet equipped to replace human judgment entirely.

A more structured approach to article quality assessment is presented by Thelwall et al. (2023) [11], who employ machine learning models to predict quality scores in alignment with the UK Research Excellence Framework. Their study demonstrates that a combination of bibliometric and textual features can effectively classify articles into quality tiers. This model, enhanced with features like argument mining and sentiment analysis, bridges the gap between citation-based predictions and qualitative review assessment.

In a similar way, Vainshtein et al. (2019) [12] introduce a content-based method for assessing scientific paper quality using corpus linguistics and domain-specific collocations. By comparing collocation usage in papers from high and low impact conferences, they show that linguistic richness and alignment with expert vocabulary can serve as strong indicators of quality. Their metric significantly outperforms traditional readability scores and baseline classifiers, highlighting the potential of domain-aware textual analysis for automated quality assessment.

Finally, datasets like PeerRead (Kang et al., 2018) [13] and APE demonstrate the growing trend of leveraging structured peer review data for training machine learning models. These datasets enable the development of models that can predict acceptance, assess clarity, and even estimate scientific impact based on review text. This shift towards structured evaluation data illustrates the ongoing evolution of peer review from a purely human-driven process to a hybrid model augmented by AI tools.

Together, these studies outline the current landscape of AI-driven evaluation in scientific peer review. They suggest that while citation metrics provide a glimpse of an article's reach, true quality assessment demands a deeper integration of argumentative analysis, peer review content, and machine learning-based predictions. As AI technologies continue to mature, their role in augmenting peer review processes is likely to expand, potentially reshaping how scientific quality is evaluated and maintained.

III. Problem Statement and Methodology

A. Problem Definition

Despite the rapid growth of preprints, there is a significant gap in the ability to efficiently assess their likelihood of being published in peer-reviewed conferences. Researchers often struggle to determine which preprints are worth further consideration due to the sheer volume of submissions and the lack of a systematic approach to predict their future success. While some AI-based approaches have attempted to evaluate the quality of articles, predicting whether a preprint will eventually lead to a formal publication remains an unsolved challenge.

This study, conducted under the guidance of a researcher as part of an academic course, aims to tackle a focused yet practical problem: **predicting the conference rank at which a preprint will likely be published**, based on its characteristics. Formally, we define the problem as a multi-class classification task:

$$f : X \to Y \tag{1}$$

where X represents the feature space derived from paper structure and metadata, and $Y = \{A^*, A, B, C, \text{unranked}\}$ represents the set of possible conference rankings. The function f maps a paper's features to its most likely ranking category.

Unlike previous efforts that focus on quality assessment or full automation of peer review, we aim to develop a predictive model that can provide valuable insights for researchers to prioritize promising preprints. Such a model could serve as a practical tool for literature reviews, guiding researchers toward preprints with higher potential for formal publication, without needing to manually sift through large volumes of unreviewed work.

To address this, the study will use arXiv as a case study, leveraging existing preprints and their eventual publication status. The core challenge is to determine which features (both from the text and metadata) are most indicative of a preprint's future acceptance, and to develop an AI model that can make reliable predictions.

B. Methodological Framework

Our methodology is designed to predict the likely conference rank of a preprint by leveraging publicly available data sources and machine learning techniques. We focus on a subset of arXiv preprints submitted between 2018 and 2023 in the artificial intelligence category. To ensure a representative yet manageable dataset, we randomly sampled 10% of the preprints from each year, resulting in a total of 6,282 documents.

For each sampled preprint, we investigate its potential publication outcome by checking whether it appears in DBLP, a curated bibliography of computer science publications. When a match is found, we extract the corresponding conference name and use the CORE 2023 ranking database to determine the rank of the venue. If the conference is ranked A*, A, B, or C, we assign the corresponding label. Conferences listed in CORE but not assigned one of these four ranks are labeled as unranked.

These labeled examples form the basis for a supervised learning task. We extract a variety of features from each preprint's metadata and content to train a classification model that predicts the publication rank.

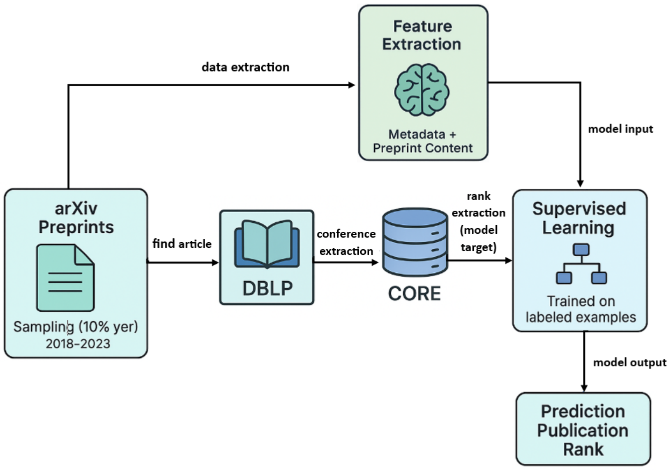

Fig. 3. Pipeline Structure of the Methodological Framework

To guide our investigation, we propose the following hypotheses based on the characteristics of scientific papers and the accuracy of the databases used for prediction.

H1: We hypothesize that various metrics, such as the number of pages, characters, authors, figures, tables, readability scores, and the presence of structured content, are positively correlated with the perceived quality, clarity, and academic importance of the paper, and that these factors collectively influence the likelihood of acceptance at high-ranked conferences.

H2: We assume that the DBLP database is up to date and accurately reflects the conference affiliations of preprints. This assumption is important as it provides reliable information to predict the future publication outcomes of preprints.

H3: We also assume that the CORE ranking database is up to date and provides accurate rankings of conferences. This assumption is crucial for using conference rankings as an indicator of the likely success of a preprint in reaching a high-quality venue.

IV. Dataset Acquisition and Preparation

In order to train and evaluate our automatic article-ranking pipeline, we first assembled a large, high-quality corpus of scientific manuscripts along with corresponding conference ranking metadata. To this end, we retrieved the PDF articles directly from arXiv and assigned each paper a conference rank based on the CORE 2023 database. This mapping relied on DBLP to determine whether an arXiv preprint had been subsequently published, and if so, to identify the corresponding conference. The identified conference was then matched to its associated rank in the CORE 2023 dataset, which we processed in CSV format. The following section details the procedures for PDF collection, conference identification via DBLP, and rank assignment using the CORE rankings.

A. arXiv PDF Retrieval

We initially considered using the official arXiv API to download a large number of articles. However, due to the volume of requests, we were blocked by CAPTCHA protections that prevented automated bulk downloading. Although the API conveniently provides direct PDF download URLs, this limitation led us to adopt a human-inspired browser automation strategy. This approach simulates manual downloads and achieves high throughput (˜464 papers/hour) while preserving download integrity.

Upon receipt of an URL of the form:

$$\textit{https://arxiv.org/abs/}\langle \textit{id} \rangle,$$

our Python script extracts the identifier (e.g., ”*2104.12345v2*”) and synthesizes the direct PDF link:

$$\textit{https://arxiv.org/pdf/}\langle \textit{id} \rangle\textit{.pdf.}$$

The script then invokes *webbrowser.open()* to navigate to this URL. After a 5-second delay (empirically chosen to accommodate network variability), *pyautogui* simulates the sequence:

1) Press `Ctrl+S`
2) Wait 0.5 seconds
3) Type `<id>.pdf`
4) Press `Enter`

which triggers the native ”Save As” dialog, specifies the filename, and confirms the download without manual intervention. All PDFs are stored under the project-relative directory *data/pdf/* which amounts to approximately 10 GB in total.

To guarantee data quality, each downloaded file is immediately checked for a minimum size (100 kB); files below this threshold—indicative of CAPTCHA pages or network errors—are automatically retried up to three times and logged if still unsuccessful. Over a continuous 14-hour run, this procedure yielded 6,282 valid PDF documents for downstream analysis.

B. Matching arXiv articles with DBLP

Initial matching based solely on arXiv titles resulted in low recall due to minor formatting differences. To improve accuracy, we adopted a three-step matching strategy. First, we queried the DBLP API by title and applied fuzzy matching (based on Levenshtein distance, called *sim* here), accepting results with a score ≥ 90. If unsuccessful, we used a SPARQL query with Finally, we performed a fallback author-based search by identifying DBLP articles with overlapping author names; if the title of a candidate matched the arXiv title with a fuzzy score of at least 70, the match was accepted.

Algorithm 1 arXiv–DBLP Matching Strategy

```
Require: arXiv title $T$, author list $A$, thresholds $\theta_{title} = 90$,
    $\theta_{author} = 70$
$(m_1, s_1) \leftarrow$ sim($T$, DBLP API titles)
if $s_1 \geq \theta_{title}$ then
    return match via API title
else
    $(m_2, s_2) \leftarrow$ sim($T$, SPARQL title search)
    if $s_2 \geq \theta_{title}$ then
        return match via SPARQL
    else
        for each candidate $c$ in DBLP API(A) do
            if authors($c$) $\cap A \neq \emptyset$ then
                $s_3 \leftarrow$ sim($T$, title of $c$)
                if $s_3 \geq \theta_{author}$ then
                    return match via author fallback
                end if
            end if
        end for
    end if
end if
return no match found
```

Once an article is successfully identified in DBLP, we can retrieve both the name of the conference and the corresponding DBLP ID. This approach improved coverage while maintaining acceptable precision.

C. Matching DBLP Conferences with CORE Rankings

First, to match DBLP conferences with CORE rankings, we download the tab-separated export *CORE2023.csv* from the official CORE portal, which contains the following columns:

$$\{\textit{ID}, \textit{Conference}, \textit{Acronym}, \textit{Source Year}, \textit{Rank}, \ldots\}.$$

Each row corresponds to a ranked conference, uniquely identified by its *ID*. To associate each conference with its corresponding DBLP source, we implemented a lightweight web scraping procedure using Python.

The script leverages the *requests* and *BeautifulSoup* libraries to iterate over each *ID* and construct the corresponding URL:

https://portal.core.edu.au/conf-ranks/$\langle$ID$\rangle$/.

It then performs HTML scraping on the retrieved page to extract the ”DBLP Source” hyperlink, located inside a `<div>` element with class `"row oddrow"`. The extracted DBLP URLs are appended as a new column, *DBLP_Source*. Next, we normalize these URLs to produce a concise DBLP key, appended as a new column, *DBLP_ID*.

To assign a CORE rank to each harvested article, we implement a two-stage matching procedure: we first attempt an exact match using the DBLP conference identifier extracted via scraping. If no match is found at this stage, we fall back to fuzzy matching between the DBLP conference name and the CORE conference name or the CORE Acronym.

Algorithm 2 Get CORE Rank from DBLP Identifier or Conference Name

```
Require: DBLP identifier $K$, conference name $L$, similarity threshold $\theta = 80$
Load CORE entries into list $C$
for each entry $c \in C$ do
    if $c$.DBLP_KEY $= K$ then
        return $c$.Rank
    else if $\max\big(\text{sim}(L, c.\text{Name}),\ \text{sim}(L, c.\text{Acronym})\big) \geq \theta$ then
        return $c$.Rank
    end if
end for
return Unranked
```

By combining these steps, we were able to establish an automated system for assigning a rank to an article within its associated conference. First, the articles were retrieved from arXiv and identified through a matching procedure with the DBLP database to determine the conference to which each article belongs. Then, each conference was mapped to its rank according to the CORE 2023 database, using an exact and fuzzy matching approach for conference names and identifiers. This process allows us to obtain an accurate ranking for each article based on the conference to which it was submitted, thus facilitating the automatic evaluation of articles.

D. Database Structure

We focus on the cs.AI category from arXiv and extract metadata using the official API. We sample up to 10% of articles from each year between 2018 and 2023. The raw data includes the basic fields from arXiv such as *title*, *summary*, *published*, *updated*, *id*, *link*, *doi*, *authors*, *categories*, *comments*, *journal_ref*, and *primary_category*. Additional features will be progressively added to this dataset. The raw data is stored in MongoDB, and the database schema was designed to support incremental feature extraction and enrichment. We created MongoDB indexes on frequently queried fields (e.g., arXiv ID, DBLP ID) to optimize retrieval performance. This structure allows us to progressively add features without re-processing the entire corpus, an important consideration when working with thousands of PDF documents.

V. Features Exploration

In this section, we detail the set of features extracted and engineered for our classification task. These features fall into three main categories: default metadata from arXiv, structural features derived from PDF content, and external metrics capturing readability and author reputation. Boxplot visualizations of these features grouped by paper rank are provided in Appendix for further inspection.

A. Default Features

We extracted the following features directly from the arXiv API:

- **Abstract size:** Measured as the number of words in the abstract. A longer abstract may indicate a more developed contribution. (Fig. 8)
- **Number of authors:** The size of the authorship team may correlate with collaboration scope or project complexity. (Fig. 12)
- **Version number:** Number of times the submission was revised. More versions may suggest higher author engagement or iteration based on feedback. (Fig. 16)

These features provide a lightweight, language-agnostic snapshot of the article before content parsing.

B. Document Features

For each downloaded PDF, we used PyMuPDF and GROBID (GeneRation Of BIbliographic Data) [14] to extract structural properties:

- **Number of characters:** Total length of the article’s full text, capturing verbosity and depth. (Fig. 9)
- **Number of pages:** A proxy for content length and article scope. (Fig. 13)
- **Number of images:** Figures can signal empirical work or conceptual clarity, often found in higher-quality submissions. (Fig. 11)
- **Number of references:** Indicates literature awareness and research maturity. (Fig. 14)

These indicators help differentiate short position papers from more comprehensive technical articles.

C. External Features

We also incorporate contextual and linguistic indicators:

- **Gunning Fog Index:** A readability score computed on the abstract text, defined as:

$$\text{GFI} = 0.4 \times \left(\frac{\text{words}}{\text{sentences}} + 100 \times \frac{\text{complex words}}{\text{words}} \right) \quad (2)$$

  where ”complex words” are those with three or more syllables. Simpler and clearer abstracts may reflect better writing quality (Fig. 15).

- **Average author h-index:** When available, we include the average h-index of the author list as a proxy for author seniority or impact. We retrieved these values using the *scholarly*, which queries Google Scholar profiles. An h-index was successfully found for at least one author in 87% of the papers. For the remaining cases, a value of 0 was assigned. (Fig. 10).

VI. Experimentation and Results

In this section, we evaluate our system's performance in predicting the publication rank of arXiv articles using a supervised classification model. We first describe the experimental setup, including the preprocessing and model configuration, followed by a presentation and interpretation of the obtained results.

A. Experimental Setup

To evaluate our pipeline, we used our filtered dataset of 6,282 scientific articles in the *cs.AI* category from arXiv, each enriched with PDF-derived statistics, metadata, and matched CORE rankings.

We selected a set of numerical features (see Section V) for training a Random Forest classifier. These include: abstract size, number of authors, number of characters, page count, image count, version number, reference count, Gunning Fog index, and average h-index. Only samples with complete data for these features were retained.

The dataset was split into training and testing subsets using an 80/20 ratio, with stratified sampling to preserve class distribution. The Random Forest model was trained using the *RandomForestClassifier* implementation from the Scikit-learn (*sklearn*) library, with 100 estimators and default hyperparameters. Specifically, the default values are: *max_depth = None*, *min_samples_split =2* ,*min_samples_leaf = 1*, *max_featurs = 'sqrt'*, and *criterion = 'gini'*. Evaluation was performed using accuracy, precision, recall, and F1-score, complemented by a confusion matrix and feature importance analysis.

B. Results

The classification model yielded weak overall performance, with limited ability to distinguish between different publication ranks. As shown in Table I, the classifier achieved a recall of 0.91 for the *UnRanked* category and an F1-score of 0.41 for the *A** category. This high score for *UnRanked* reflects the model's tendency to default to majority-class predictions, rather than a genuine ability to detect unranked articles. In contrast, most ranked articles were misclassified as *UnRanked*, as illustrated in the confusion matrix (Figure 4). Interestingly, the model showed some capacity to distinguish top-tier papers (*A**) from *UnRanked*, but failed almost entirely to identify intermediate-ranked classes (*A*, *B*, and *C*). This suggests that while extreme classes may exhibit distinguishable patterns, the feature set does not capture enough signal to separate finer-grained tiers.

TABLE I
Classification Report by Rank

Rank	Precision	Recall	F1-score	Support
A*	0.45	0.37	0.41	210
A	0.20	0.02	0.04	97
B	0.44	0.04	0.07	200
C	0.00	0.00	0.00	54
UnRanked	0.58	0.91	0.71	639

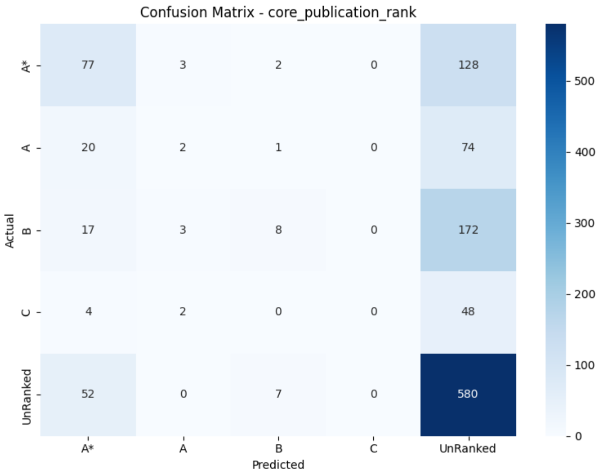

Fig. 4. Confusion matrix showing prediction versus actual rank labels.

Figure 5 shows the relative importance of each feature in this specific classification model. The number of characters in the full text emerged as the most influential feature, followed by abstract size, Gunning Fog index, and average author h-index. Reference count also contributed moderately, while version number and number of authors were among the least informative. Despite these insights, the top features appear to capture general writing style and document structure rather than providing reliable signals of publication venue quality.

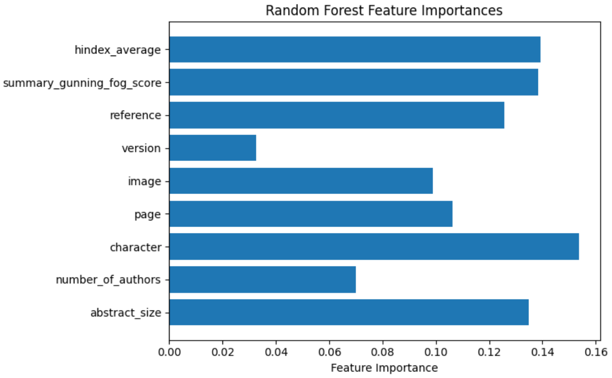

Fig. 5. Random Forest feature importance analysis.

To further evaluate model consistency, we performed 10-fold cross-validation. The model achieved an average accuracy of 53.9% with a standard deviation of 1.0%. However, macro-averaged precision, recall, and F1-score were considerably lower (26.6%, 24.4%, and 21.3%, respectively), indicating poor performance across the majority of classes. This confirms that the model predominantly learns to classify the majority class (*UnRanked*) and fails to distinguish reliably among ranked categories. The observed accuracy is therefore inflated by class imbalance and should not be interpreted as an indicator of generalization.

Overall, these results highlight the limitations of using surface-level structural and readability features to predict publication venue rank, and motivate the need for deeper semantic or citation-based features in future iterations.

C. Extended Experimentation

To further evaluate our model's capacity to distinguish between different quality tiers of scientific publications, we conducted a second experiment that excluded the overrepresented *UnRanked* category. We made this decision after discovering a bias in our dataset: several papers labeled as *UnRanked* between 2018 and 2023 had in fact been published in ranked conferences after 2023, outside the scope of our ground-truth collection. This temporal limitation in the CORE ranking database introduced label noise, particularly affecting the *UnRanked* class. Excluding these samples allowed us to reassess the classifier's ability to distinguish between genuinely ranked articles (A*, A, B, and C) without the distortion caused by incomplete future publication data. This modification aimed to assess whether the classifier could more effectively differentiate between articles with assigned CORE rankings when the dominant class was removed. For this follow-up experiment, we maintained the same feature set and Random Forest classifier configuration as in our initial experiment (100 estimators with default hyperparameters). All other aspects of the methodology remained consistent, including the preprocessing steps, feature selection, and evaluation metrics. We employed both a single train-test split (80/20 ratio with stratified sampling) and 10-fold cross-validation to ensure robust performance assessment.

1) 80/20 ratio with stratified sampling:

The removal of the *UnRanked* class altered the classification dynamics significantly. Table II presents the performance metrics for the single train-test split configuration.

TABLE II
CLASSIFICATION REPORT FOR RANKED PAPERS ONLY

Rank	Precision	Recall	F1-score	Support
A*	0.51	0.67	0.58	226
A	0.15	0.04	0.06	102
B	0.45	0.57	0.50	209
C	0.00	0.00	0.00	56

The confusion matrix (Figure 6) provides further insight into the model's classification behavior.

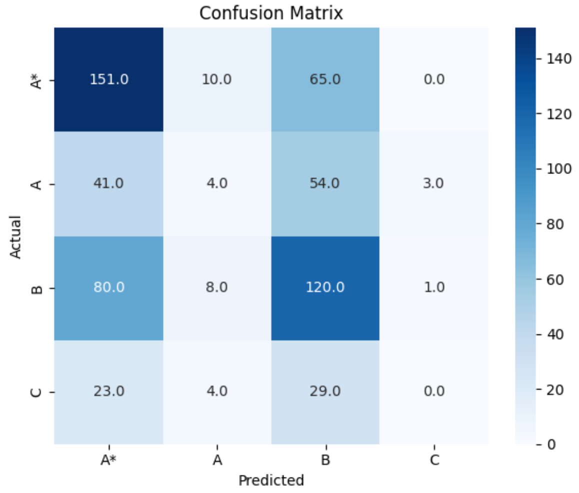

Fig. 6. Confusion matrix for ranked papers only.

The classifier achieved an overall accuracy of 46.4%, which represents a modest improvement over the multi-class performance observed in our first experiment when considering only the ranked categories. Notably, the model maintained moderate success in identifying A* and B-ranked papers, with recall values of 0.67 and 0.57 respectively. However, it continued to struggle with A-ranked papers (0.04 recall) and completely failed to identify C-ranked papers (0.00 recall).

In comparison to our initial experiment, where the model primarily defaulted to the majority class (*UnRanked*), the refined model showed improved capability to distinguish between certain ranked categories, particularly A* and B. Nevertheless, the challenges in identifying intermediate and lower-ranked papers persisted.

2) Cross-Validation:

To ensure the robustness of our findings, we performed 10-fold cross-validation on the ranked papers dataset. The model demonstrated fairly consistent performance across all folds, with an average accuracy of 50.4% and a standard deviation of 2.9%. Table III summarizes the performance metrics averaged across all folds.

TABLE III
AVERAGE PERFORMANCE METRICS ACROSS 10 FOLDS

Rank	Precision	Recall	F1-score	Support
A*	0.54	0.73	0.61	113
A	0.21	0.05	0.08	51
B	0.49	0.62	0.54	104
C	0.15	0.02	0.03	28

The cross-validation analysis revealed consistent patterns across folds:

- The highest accuracy was achieved in Fold 2 (56.9%), while the lowest was in Fold 5 (47.3%)
- A* papers were consistently classified with reasonable accuracy (F1-scores ranging from 0.58 to 0.70)
- B-ranked papers showed moderate consistency in classification (F1-scores between 0.51 and 0.62)
- A and C-ranked papers remained difficult to classify correctly across all folds

The average confusion matrix across all folds further illustrates these patterns:

TABLE IV
AVERAGE CONFUSION MATRIX ACROSS 10 FOLDS

	Predicted A*	Predicted A	Predicted B	Predicted C
Actual A*	82.1	3.3	27.0	0.6
Actual A	24.7	2.5	23.4	0.5
Actual B	35.5	4.1	64.2	0.5
Actual C	10.1	1.0	16.1	0.6

This matrix reveals that A* papers were most frequently correctly classified, while A-ranked papers were rarely identified correctly. The model showed a tendency to misclassify A and C ranked papers as either A* or B ranked, suggesting difficulty in capturing the more nuanced differences between publication tiers.

3) Feature Importance Analysis:
The feature importance analysis for this refined experiment remained largely consistent with our initial findings. As illustrated in Figure 7, character count continued to be the most influential feature, with a substantial margin over other features.

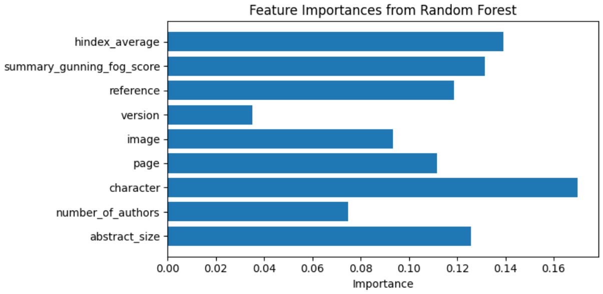

Fig. 7. Random Forest feature importance analysis for ranked papers only.

Abstract size, h-index average, and Gunning Fog score also maintained their relative importance, reinforcing the observation that document structure and author reputation metrics provide some signal for distinguishing publication quality. Number of authors and version number remained among the least informative features, suggesting limited correlation with publication venue ranking in our dataset.
In comparison to our initial feature importance analysis, the relative ordering of features remained largely stable, indicating that the exclusion of *UnRanked* papers did not substantially alter the feature relevance patterns identified by the Random Forest classifier.

D. Limitations

Our approach faces several limitations:

- **Content-agnostic classification:** Our features capture paper structure but not semantic content, potentially missing breakthrough ideas presented in unconventional formats.
- **Domain specificity:** The current model is trained and validated on computer science (cs.AI) papers; different disciplines may exhibit different structural patterns.
- **Circular reinforcement:** If structural features indeed predict rankings, using them to filter submissions could reinforce existing patterns rather than identifying innovative but structurally atypical work.
- **Ground truth quality:** CORE rankings themselves represent subjective assessments that may not perfectly capture research quality or impact.

E. Alternative Approaches and Combined Metrics

In addition to our primary classification model, we explored several alternative approaches based on Large Language Models (LLMs) to evaluate scientific paper quality. We conducted a systematic investigation to determine whether LLMs could effectively differentiate between high-quality (A*) and lower-ranked (C) papers in computer science.
1) LLM Prompting Experiments: We hypothesized that if LLMs could reliably distinguish between papers at opposite ends of the quality spectrum (A* vs. C), they could potentially serve as valuable tools for automated paper evaluation. We designed seven distinct prompting strategies using ChatGPT(gpt 4o mini) to test this hypothesis, each approaching the evaluation task from a different angle:

- **Prompt 1: Tagged Review with Scores** - We prompted the model to write academic-style reviews with specific tags ([MOTIVATION], [SUBSTANCE], [ORIGINALITY], etc.) followed by numerical scoring on five dimensions (Soundness/Correctness, Originality, Clarity, Relevance, and Methodology) using a 1-5 scale.
- **Prompt 2: Untagged Review with Scores** - Similar to Prompt 1 but without the structured tags, requesting a free-form academic review followed by scoring on the same five dimensions.
- **Prompt 3: Untagged Review with Single Score** - Requesting a free-form academic review followed by a single overall score from 1-10.
- **Prompt 4: Direct Scoring** - Requesting only a numerical score from 1-10 without any review text.
- **Prompt 5: Review with Editorial Decision** - Requesting a free-form academic review followed by a categorical prediction (Accept, Reject, Minor Revision, Major Revision).
- **Prompt 6: Direct Editorial Decision** - Requesting only an editorial decision without any review.
- **Prompt 7: Percentile Estimation** - Requesting the model to estimate the paper's percentile ranking among all submissions.

For each prompting strategy, we evaluated three A* papers and three C papers to assess the model's ability to differentiate between quality tiers. The specific prompts used in our experiments are included in the Appendix.
2) LLM Evaluation Results: Our experimental results revealed a striking pattern: across all seven prompting strategies, the LLM consistently failed to differentiate

between papers of different quality rankings. The results for each prompting strategy are summarized below:

- **Prompt 1 (Tagged Review with Scores):** For both A* and C papers, the model assigned remarkably similar scores across all five dimensions. A* papers received scores of SOUNDNESS_CORRECTNESS: 4, ORIGINALITY: 5, CLARITY: 4, RELEVANCE: 5, METHODOLOGY: 4-5. C papers received nearly identical scores: SOUNDNESS_CORRECTNESS: 4, ORIGINALITY: 4-5, CLARITY: 3-4, RELEVANCE: 5, METHODOLOGY: 4.
- **Prompt 2 (Untagged Review with Scores):** The results were virtually identical to Prompt 1, with no meaningful differentiation between A* and C papers.
- **Prompt 3 (Review with Single Score):** A* papers received scores of 7-8 out of 10, while C papers uniformly received scores of 8 out of 10, failing to reflect their lower tier ranking.
- **Prompt 4 (Direct Scoring):** Both A* and C papers received uniform scores of 8 out of 10, showing no discrimination between quality tiers.
- **Prompt 5 (Review with Editorial Decision):** A* papers received decisions of ”Minor Revision” (2 papers) and ”Major Revision” (1 paper), while C papers received ”Minor Revision” (1 paper) and ”Major Revision” (2 papers), showing no clear pattern reflecting their ranking difference.
- **Prompt 6 (Direct Editorial Decision):** Similar to Prompt 5, no consistent pattern emerged that could differentiate between quality tiers.
- **Prompt 7 (Percentile Estimation):** A* papers were estimated to be in the 10-15th percentile range, while C papers were estimated to be in the 10-15th percentile range as well, showing no differentiation despite their actual ranking disparity.

TABLE V
LLM EVALUATION RESULTS FOR DIFFERENT PROMPTS

Prompt Approach	**A* Papers**	**C Papers**
Multi-dimensional Scores	High scores (4-5)	High scores (4-5)
Single Score/10	7-8	8
Direct Scoring/10	8	8
Editorial Decision	Minor/Major Revision	Minor/Major Revision
Percentile	10-15	10-15

These results suggest that current LLMs, despite their sophistication in generating plausible-sounding reviews, lack the ability to make meaningful quality distinctions between papers of different ranks. This finding has significant implications for the potential use of LLMs in scientific evaluation and peer review processes.

VII. DISCUSSION

Our experimental results reveal significant limitations in the automated prediction of conference rankings using both traditional machine learning and LLM-based approaches. Contrary to our initial hypotheses, we found that neither structural features nor language model evaluations could reliably distinguish between papers of different quality tiers.

A. Limitations of Structural Features

The poor performance of our Random Forest classifier highlights several important limitations:

- **Surface-level indicators:** The structural and metadata features we extracted (reference count, page count, readability scores, etc.) appear to be insufficient proxies for research quality or impact.
- **Quality complexity:** Scientific quality is fundamentally multidimensional and context-dependent, involving factors like methodological rigor, theoretical innovation, and practical significance that may not be captured by structural features alone.
- **Domain variation:** Even within the cs.AI category, different sub-domains may have distinct conventions and expectations regarding paper structure and presentation.
- **Ranking subjectivity:** Conference rankings themselves represent subjective assessments that may not perfectly align with objective quality metrics.

B. Limitations of LLM-Based Evaluation

The inability of Large Language Models to differentiate between papers of different ranks raises important concerns about their current capabilities for scientific evaluation:

- **Positive bias:** Our experiments revealed a consistent tendency for LLMs to generate overly positive assessments, assigning high scores even to papers from lower-ranked venues.
- **Lack of critical evaluation:** The models appear to struggle with identifying substantive methodological or theoretical weaknesses—precisely the discriminative qualities needed for accurate ranking.
- **Limited domain expertise:** Despite their broad training, current LLMs may lack the specialized knowledge required to evaluate cutting-edge research in specific domains.
- **Superficial analysis:** The models may focus on surface-level features like writing style and presentation rather than deeper qualities of scientific contribution.

These findings align with recent work by Luo et al. [8], who found that LLMs can generate syntactically fluent but often superficial reviews that fail to capture fine-grained scientific critiques.

C. Ethical Considerations

Our research raises important ethical questions about automated evaluation systems in scientific publishing:

- **False authority:** The convincing but potentially unreliable assessments generated by LLMs could be mistakenly given undue weight in academic decision-making.
- **Algorithmic bias:** Models trained on existing publication patterns may perpetuate historical biases in the scientific community.

- **Transparency requirements:** The limitations of automated evaluation tools should be clearly communicated to all stakeholders in the publishing process.
- **Human oversight:** Our results strongly suggest that human expertise remains irreplaceable in scientific evaluation, particularly for assessing novelty and significance.

D. Future Work

Despite these limitations, several promising directions for future research emerge:

- **Semantic content analysis:** Moving beyond structural features to incorporate other features with deeper semantic understanding of paper content.
- **Expert-guided evaluation:** Developing hybrid systems that combine automated tools with human expertise in a complementary manner.
- **Citation and impact integration:** Incorporating post-publication metrics like citation patterns to refine predictive models.
- **Specialized domain knowledge:** Training or fine-tuning models on domain-specific literature to improve their evaluative capabilities.

VIII. Conclusion

In this paper, we investigated the challenging problem of automatically predicting conference rankings for scientific papers using both traditional machine learning on structural features and LLM-based evaluation approaches. Contrary to our initial expectations, our experimental results demonstrate that neither approach can reliably distinguish between papers published in venues of different ranks. The Random Forest classifier showed poor performance across most ranking categories, while LLM evaluations consistently failed to differentiate between A* and C papers across multiple prompting strategies.

These findings highlight the complex nature of scientific quality assessment and the limitations of current computational approaches. Quality in scientific research encompasses numerous dimensions including methodological rigor, theoretical innovation, empirical validation, and potential impact aspects that may not be reliably captured by structural features or current language models.

Our work makes several contributions to the field: (1) it provides a systematic evaluation of structural features for quality prediction; (2) it offers empirical evidence of the limitations of current LLMs for scientific evaluation; and (3) it identifies important directions for future research in automated scientific assessment.

While automation tools may eventually play a supportive role in the scientific publishing ecosystem, our research suggests that human expertise remains irreplaceable for meaningful quality evaluation. The complex, multifaceted nature of scientific excellence continues to resist simple quantification or algorithmic judgment, reinforcing the ongoing importance of peer review as a cornerstone of scientific progress. Moving forward, research should focus on developing tools that complement rather than replace human judgment, identifying specific assessment tasks where automation can provide genuine assistance, and maintaining a critical perspective on the limitations of algorithmic evaluation in scientific contexts.

Keywords

Scientific Paper Ranking, LLM, Peer Review Automation, arXiv Metadata, CORE Conference Rankings, Artificial Intelligence in Scientific Evaluation

Code Availability

All source code developed for data collection, preprocessing, feature extraction, and classification is publicly available on the following GitHub repository:
https://github.com/PLONTZNathanTN/pidr-article-rank-prediction
This repository contains all the scripts and documentation necessary to reproduce the experiments and results presented in this paper.

Acknowledgments

This project was carried out under the supervision of Professor Jérôme François, whose guidance helped define the research scope and methodology. We also acknowledge the training provided by the faculty of Télécom Nancy, which enabled the technical development of this work. Portions of this manuscript were enriched using generative AI tools (e.g., ChatGPT) for language improvement and content development. The authors edited and validated all AI-generated content to ensure accuracy and originality.

Appendix

Below we provide the exact prompts used in our LLM evaluation experiments:

A. Prompt 1: Tagged Review with Scores

Review:
You are a professor in computer science, machine learning and artificial intelligence. Write an academic-style review of the following scientific article using the following tags to structure your response: [MOTIVATION], [SUBSTANCE], [ORIGINALITY], [SOUNDNESS], [CLARITY], [STRENGTHS], [WEAKNESSES]. Each tag should introduce a short paragraph focused on that specific aspect. The review must not exceed 700 tokens, but it is not necessary to reach this limit if the content does not require it. Maintain a formal and objective tone suitable for an academic setting.
Score:
Based on the attached scientific article and the review text, assign each scores from 1 to 5 (higher means better). Output format (no deviations, no extra words, no punctuation other than shown below, no field renaming, no explanations):
SOUNDNESS_CORRECTNESS: x, ORIGINALITY: x, CLARITY: x, RELEVANCE: x, METHODOLOGY: x

B. Prompt 2: Untagged Review with Scores

Review:
You are a professor in computer science, machine learning and artificial intelligence. Write an academic-style review of the following scientific article. The review should include a concise summary of the paper's objectives, methods, results, and conclusions. Critically evaluate the strengths and weaknesses of the study, discuss its originality and contribution to the field, and suggest possible improvements or future research directions. Maintain a formal and objective tone suitable for an academic setting. The review should not exceed 700 tokens, and it is not necessary to reach this limit if the content does not require it.
Score:
Based on the attached scientific article and the review text, assign each scores from 1 to 5 (higher means better). Output format (no deviations, no extra words, no punctuation other than shown below, no field renaming, no explanations):
SOUNDNESS_CORRECTNESS: x, ORIGINALITY: x, CLARITY: x, RELEVANCE: x, METHODOLOGY: x

C. Prompt 3: Untagged Review with Single Score

Review:
You are a professor in computer science, machine learning and artificial intelligence. Write an academic-style review of the following scientific article. The review should include a concise summary of the paper's objectives, methods, results, and conclusions. Critically evaluate the strengths and weaknesses of the study, discuss its originality and contribution to the field, and suggest possible improvements or future research directions. Maintain a formal and objective tone suitable for an academic setting. The review should not exceed 700 tokens, and it is not necessary to reach this limit if the content does not require it.
Score:
Based on the attached scientific article and the review text assign an overall evaluation score from 1 to 10 (10 being the highest). Only output the score, nothing else.

D. Prompt 4: Direct Scoring

Score:
You are a professor in computer science, machine learning and artificial intelligence. Based on the attached scientific article assign an overall evaluation score from 1 to 10 (10 being the highest). Only output the score, nothing else.

E. Prompt 5: Review with Editorial Decision

Review:
You are a professor in computer science, machine learning and artificial intelligence. Write an academic-style review of the following scientific article. The review should include a concise summary of the paper's objectives, methods, results, and conclusions. Critically evaluate the strengths and weaknesses of the study, discuss its originality and contribution to the field, and suggest possible improvements or future research directions. Maintain a formal and objective tone suitable for an academic setting. The review should not exceed 700 tokens, and it is not necessary to reach this limit if the content does not require it.
Prediction:
Based on the attached scientific article and the review text, predict the likely editorial decision for an academic conference. Choose only one of the following options and output only that decision: Accept, Reject, Minor Revision, Major Revision

F. Prompt 6: Direct Editorial Decision

Prediction:
Based on the attached scientific article predict the likely editorial decision for an academic conference. Choose only one of the following options and output only that decision: Accept, Reject, Minor Revision, Major Revision

G. Prompt 7: Percentile Estimation

Top Percentage:
Based on the attached scientific article, estimate in which percentile this paper would likely rank among all submissions. Output only the estimated percentile nothing else.

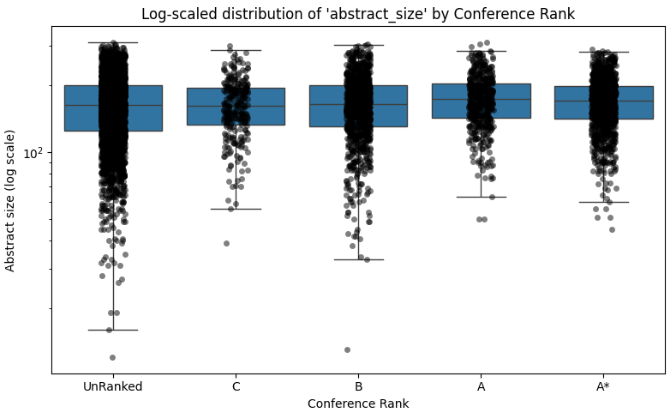

Fig. 8. Abstract size by rank

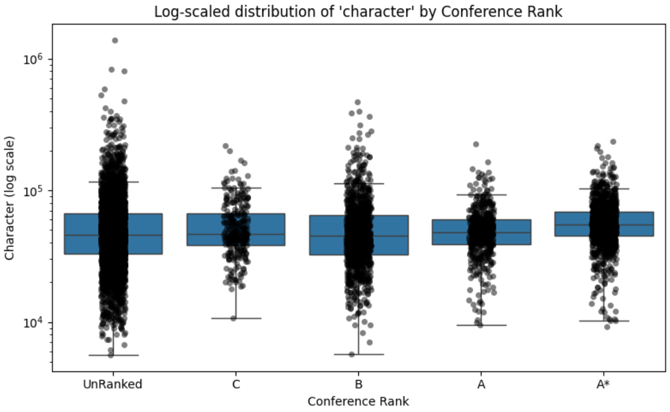

Fig. 9. Number of characters by rank

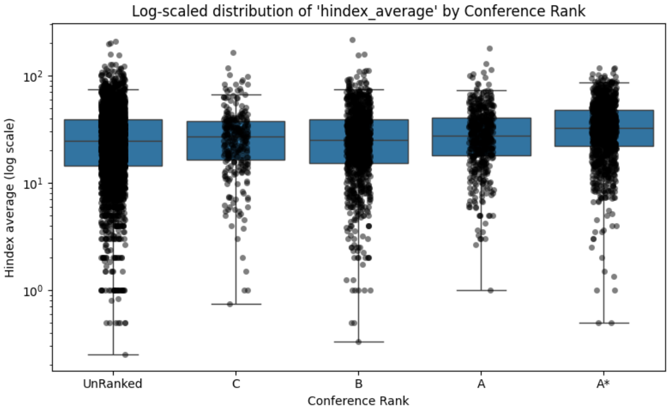

Fig. 10. Average hindex of authors by rank

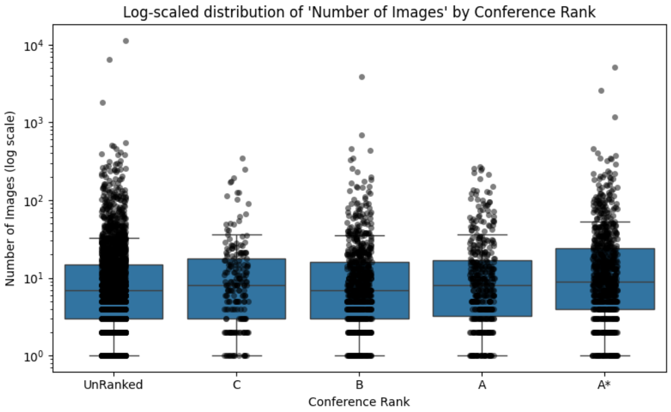

Fig. 11. Number of images by rank

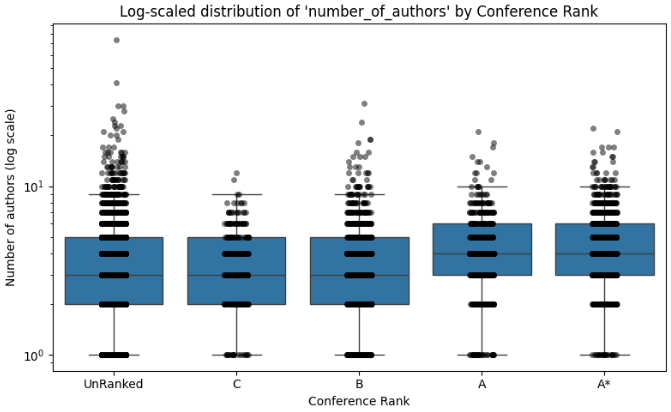

Fig. 12. Number of authors by rank

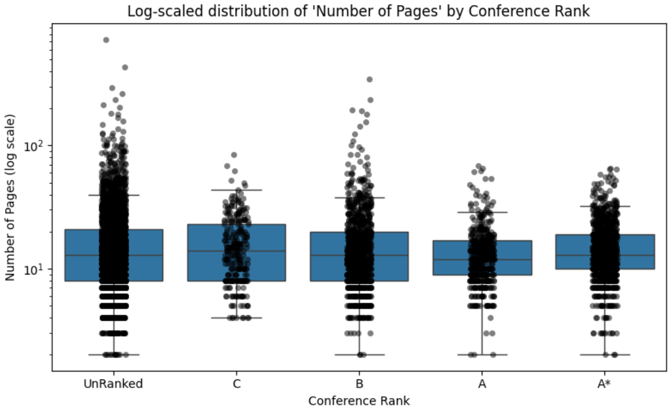

Fig. 13. Number of pages by rank

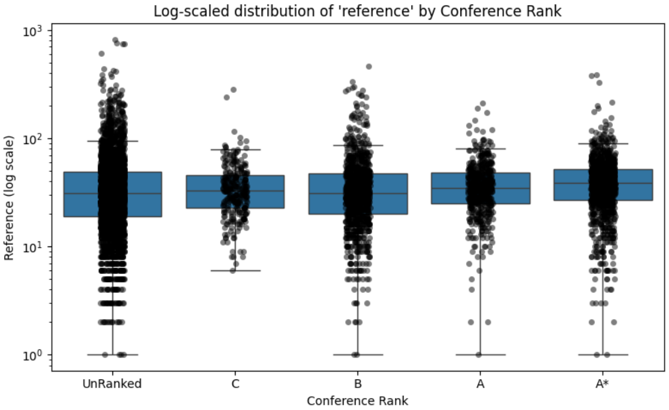

Fig. 14. Number of references by rank

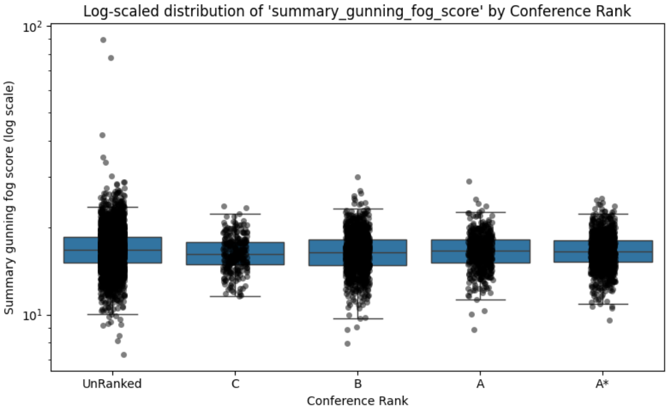

Fig. 15. Gunning fog score of the abstract by rank

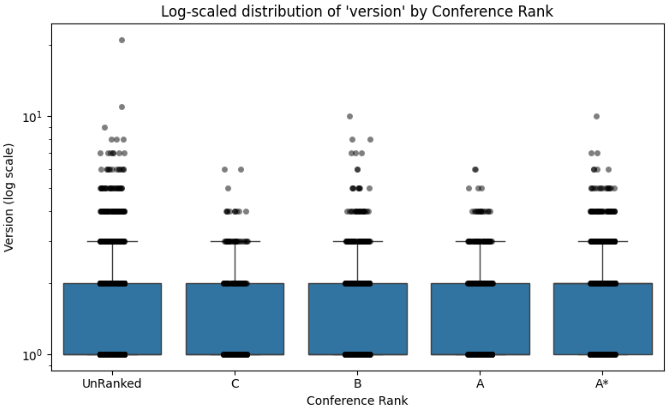

Fig. 16. Number of versions by rank

References

[1] Arxiv, "2021 statistics by area," 2021. Accessed: 2025-05-12.

[2] D. Curcic, "Nombre d'articles académiques publiés par an." https://wordsrated.com/number-of-academic-papers-published-per-year/, 2023. Consulté le 12 mai 2025.

[3] M. Szklo, "Quality of scientific articles," *Revista de saúde pública*, vol. 40 Spec no., pp. 30–5, 09 2006.

[4] T. Yu, G. Yu, P.-Y. Li, and L. Wang, "Citation impact prediction for scientific papers using stepwise regression analysis," *Scientometrics*, vol. 101, 11 2014.

[5] L. Cheng, L. Bing, Q. Yu, W. Lu, and L. Si, "Ape: Argument pair extraction from peer review and rebuttal via multi-task learning," pp. 7000–7011, 01 2020.

[6] S. Singh, M. Singh, and P. Goyal, "Compare: A taxonomy and dataset of comparison discussions in peer reviews," pp. 238–241, 09 2021.

[7] X. Hua, M. Nikolov, N. Badugu, and L. Wang, "Argument mining for understanding peer reviews," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 2131–2137, Association for Computational Linguistics, June 2019.

[8] S. Wu, X. Ma, D. Luo, L. Li, X. Shi, X. Chang, X. Lin, R. Luo, C. Pei, Z. Zhao, and J. Gong, "Automated review generation method based on large language models," 07 2024.

[9] R. Zhou, L. Chen, and K. Yu, "Is LLM a reliable reviewer? a comprehensive evaluation of LLM on automatic paper reviewing tasks," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, eds.), (Torino, Italia), pp. 9340–9351, ELRA and ICCL, May 2024.

[10] R. Liu and N. Shah, "Reviewergpt? an exploratory study on using large language models for paper reviewing," 06 2023.

[11] M. Thelwall, K. Kousha, P. Wilson, M. Makita, M. Abdoli, E. Stuart, J. Levitt, P. Knoth, and M. Cancellieri, "Predicting article quality scores with machine learning: The uk research excellence framework," *Quantitative Science Studies*, vol. 4, pp. 1–33, 04 2023.

[12] R. Vainshtein, G. Katz, B. Shapira, and L. Rokach, "Assessing the quality of scientific papers," 08 2019.

[13] D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. Hovy, and R. Schwartz, "A dataset of peer reviews (PeerRead): Collection, insights and NLP applications," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji, and A. Stent, eds.), (New Orleans, Louisiana), pp. 1647–1661, Association for Computational Linguistics, June 2018.

[14] "Grobid." https://github.com/kermitt2/grobid, 2008–2025.