UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN
Mathematics and Computer Science

Cycle XXXVIII

TITLE OF THE Ph.D. THESIS

Semantic Reviewer Matching at Conference Scale with LLM-Generated Keywords and Knowledge-Graph Profiles

Scientific Disciplinary Sector(s)

INF/01

Ph.D. Student:	Farid Bagheri
Supervisor	Prof.Diego Angelo Gaetano Reforgiato Recupero

Final exam. Academic Year 2024/2025
Thesis defence session: February 2026

Statement of Authorship

I declare that this thesis entitled "Semantic Reviewer Matching at Conference Scale with LLM-Generated Keywords and Knowledge-Graph Profiles" and the work presented in it are my own. I confirm that:

- this work was done while in candidature for this PhD degree;
- when I consulted the work published by others, this is always clearly attributed;
- when I quoted the work of others, the source is always given;
- I have acknowledged all main sources of help;
- with the exception of the above references, this thesis is entirely my own work;
- appropriate ethics guidelines were followed to conduct this research;
- for work done jointly with others, my contribution is clearly specified.

Abstract

The rapid increase in the number of submissions to scientific conferences and journals has made the assignment of qualified reviewers a critical and challenging task. Manual assignment and simple keyword-based methods do not scale well and often fail to capture the semantic diversity and interdisciplinarity of modern research. This thesis addresses the reviewer assignment problem by investigating content-based methods that improve how the topical relevance between papers and reviewers is estimated, while remaining interpretable and applicable at conference scale.

The thesis makes three main contributions. First, it introduces and analyzes a large real-world dataset of conference submissions and reviewer profiles in the areas of Computer Science and the Semantic Web, enabling realistic and reproducible evaluation of reviewer matching methods. Second, it systematically studies different ways of representing papers and reviewers using textual information, structured knowledge, and automatically extracted keywords. In particular, it compares traditional text-based similarity approaches with knowledge-graph–enriched representations and keyword-based profiles generated with large language models, showing how these complementary signals improve the estimation of reviewer expertise. Third, the thesis proposes an automated framework in which large language models assist the reviewer assignment process by producing concise, human-readable keyword representations that support transparent and effective matching.

Extensive experiments demonstrate that combining lexical overlap and semantic similarity leads to more robust reviewer recommendations under realistic assignment constraints, such as limited reviewer capacity and conflict avoidance. The results show that large language models and knowledge-based representations can significantly enhance reviewer matching while preserving interpretability and auditability. Overall, this work provides practical and scalable methods that support program chairs and editors in assigning reviewers more accurately and fairly in modern peer-review workflows.

Dissemination

The research that contributed to the content of this Ph.D. thesis has resulted in *six* published or accepted papers in international journals, conferences, and workshop proceedings. I would like to sincerely thank my co-authors for their valuable contributions. Such gratitude is reflected in the adoption of the scientific "We" throughout this thesis.

I would like to clarify my contribution to the presented works. I envisioned the research directions explored in this thesis and completed the majority of the work. I designed the approaches, collected and processed the datasets, conducted the data analysis, implemented the corresponding scripts, and wrote the papers for submission. I also managed the peer-review process and revised the manuscripts accordingly. Throughout all stages, I collaborated closely with my co-authors, who provided guidance, feedback, and technical discussion that helped refine the research. Furthermore, I was responsible for the presentation of the accepted works at international conferences and workshops.

Peer-reviewed Publications in Journals and Conference Proceedings

i. Bagheri, F., Buscaldi, D., & Reforgiato Recupero, D. (2026). *Assigning Reviewers with LLM Agents: Automating Expertise Matching in Peer Review. Scientometrics.* [Accepted].

ii. Bagheri, F., Buscaldi, D., & Reforgiato Recupero, D. (2026). *Reviewer Matching with LLMs: A Keyword-Driven, Interpretable Framework.* In: *Proceedings of the European Conference on Information Retrieval (ECIR 2026).* [Accepted].

iii. Bagheri, F., Buscaldi, D., & Reforgiato Recupero, D. (2025). *Leveraging Knowledge Graphs and LLMs for Content-Based Reviewer Assignment. Intelligent Information Systems.* [In Press].

iv. Bagheri, F., Buscaldi, D., & Reforgiato Recupero, D. (2025). *Leveraging LLMs for Content-Based Reviewer Assignment in Computer Science Conferences.* In: *Proceedings of the GOLE–EGC Workshop*, Strasbourg, France. [Accepted].

v. Bagheri, F., Buscaldi, D., & Reforgiato Recupero, D. (2024). *A Study on Content-Based Reviewer Assignment in the Semantic Web and Computer Science Domains. Computación y Sistemas*, 28(4), 2024.

vi. Bagheri, F., Reforgiato Recupero, D., & Sirnes, E. (2023). *Leveraging Return Prediction Approaches for Improved Value-at-Risk Estimation. Data (MDPI)*, 8(9), 2023. https://doi.org/10.3390/data8090123

Conference Presentations

- **Workshop GOLE–EGC 2025**, co-located with the 25th *Extraction et Gestion des Connaissances (EGC 2025)*, Strasbourg, France, January 27–31, 2025 — Oral presentation titled *Leveraging LLMs for Content-Based Reviewer Assignment in Computer Science Conferences.*

- **9th International Symposium on Language & Knowledge Engineering (LKE 2024)**, Technological University Dublin, Ireland, June 4–6, 2024 — Oral presentation titled *A Study on Content-Based Reviewer Assignment in the Semantic Web and Computer Science Domains.*

Acknowledgements

I would like to express my deepest gratitude to all the people who supported me throughout my PhD journey, both within and outside the research environment.

First and foremost, I would like to sincerely thank my main supervisor, **Prof. Diego Angelo Gaetano Reforgiato Recupero**, for his invaluable guidance, trust, and continuous encouragement during these years. His support has been fundamental in shaping my academic path and personal growth as a researcher.

I am also deeply grateful to **Prof. Davide Buscaldi** at *Université Sorbonne Paris Nord*, *Laboratoire d'Informatique de Paris Nord (LIPN)*, for his constant support, insightful advice, and for welcoming me into his research group. His mentorship and encouragement have been crucial for my development and for many of the achievements of this work.

I would like to thank all my colleagues and friends in both Cagliari and Paris, with whom I shared inspiring discussions, challenges, and many memorable moments inside and outside the lab. Your presence made this journey not only productive but also enjoyable.

A heartfelt thanks goes to my **parents** and my **brother, Farshid**, for their unconditional love, patience, and support in every step of my life. Without your encouragement, none of this would have been possible.

Finally, my deepest appreciation goes to my **wife, Merve**, for her endless love, understanding, and strength. Your support has illuminated my most difficult moments and motivated me to pursue my ambitions with determination. Thank you for always believing in me and celebrating every small success as if it were your own.

Contents

Statement of Authorship **I**

Abstract **II**

Dissemination **IV**

Acknowledgements **VII**

List of Figures **XIII**

List of Tables **XV**

1 Introduction **1**
1.1 Motivation and Open Issues 1
1.2 Research Questions 2
1.3 Methodological Overview 2
1.4 Data and Evaluation Protocol 3
1.5 Contributions 3
1.6 Ethical Scope and Roadmap 4

2 Related Work **7**

3 Problem Definition and Task Formulation **11**
3.1 Reviewer Assignment Problem 11
3.2 Core Objectives of the Task 12
3.2.1 Semantic Feature Extraction 12
3.2.2 Similarity Computation and Reviewer Matching 12
3.2.3 Assignment under Constraints 13
3.3 Evaluation and Ranking 13

4 Reviewer Assignment Pipeline **15**
4.1 Pipeline Overview 15
4.2 Data Collection 16
4.2.1 Authors' Dataset 17
4.2.2 Reviewers' Dataset 17

4.2.3 Integration, Format, and Availability . . . 17
4.3 Preprocessing . . . 18
4.3.1 Semantic Triple Extraction (OpenIE) . . . 19
4.3.2 Graph-Based Linguistic Named Entity Recognition (GLiNER) 20
4.3.3 CSO Classifier . . . 21
4.3.4 LLM-Based Keyword Extraction . . . 22
4.3.5 LLM Agentic pipeline . . . 24
4.3.6 Knowledge Graph Construction . . . 24
4.3.7 Similarity Computation and Reviewer Recommendation . . . 26
4.3.7.1 Cosine Similarity . . . 26
4.3.7.2 Jaccard Similarity . . . 27
4.3.7.3 Combined Similarity . . . 27
4.3.7.4 Dot Product Similarity . . . 27

5 Evaluation **29**
5.1 Overview . . . 29
5.2 Evaluation Metrics . . . 29
5.2.1 Mean Reciprocal Rank (MRR) . . . 30
5.2.2 Precision at k (P@k) . . . 30
5.2.3 Mean Average Precision (MAP) . . . 31
5.3 Workload and Fairness Metrics . . . 31
5.3.1 Per-Conference Average Workload . . . 32
5.3.2 Macro-Average Workload . . . 32
5.3.3 Pooled Mean Workload . . . 32
5.3.4 Gini Coefficient (Workload Inequality) . . . 33
5.4 Experiments . . . 33

6 Results **35**
6.1 Overview . . . 35
6.2 Results Based on Paper Titles . . . 36
6.2.1 Classical content-based baselines (Title Only) . . . 36
6.2.2 LLM-Based Approach (Title Only) . . . 38
6.2.3 LLM Agentic Pipeline (Title Only) . . . 39
6.3 Results Based on Summary of abstracts . . . 41
6.3.1 Classical Baselines (Summary of Abstracts) . . . 41
6.3.2 LLM-Based Approach (Summary of Abstracts) . . . 43
6.3.3 LLM Agentic Pipeline (Summary of Abstracts) . . . 44
6.4 Results Based on Abstracts . . . 46
6.4.1 Classical Baselines (Abstracts) . . . 46
6.4.2 Knowledge Graphs (Typed Relations from Abstracts) . . . 48
6.4.3 Knowledge Graphs (`related_to` Edge from Abstracts) . . . 49
6.4.4 LLM-Based Approach (Abstracts) . . . 51
6.4.5 LLM Agentic Pipeline (Abstracts) . . . 52

6.5 Assignment Realism and Fairness 54
6.6 Statistical Reliability and Robustness 56

7 Conclusion **59**
7.1 Summary of Contributions and Findings 59
7.2 Limitations 60
7.3 Future Directions 60
7.4 Closing Remarks 61

Bibliography **63**

List of Figures

4.1 Reviewer Assignment Pipeline. 16

6.1 Effect of assignment constraints on classical title-only baselines (constrained − unconstrained). Improvements are shown across standard metrics: MRR, MAP, and P@3. 38

6.2 Titles-only reviewer matching. Early precision (P@3) and depth (MRR, MAP) for classical baselines (unconstrained vs. constrained), the best single-LLM setting (GPT-4o-mini, Jaccard), and the agentic pipeline (Cosine-Jaccard Average). The agent improves top-k precision while dot-product (constrained, mean) leads MAP. 41

6.3 Effect of assignment constraints on classical baselines using summary of abstracts (constrained − unconstrained). Values show absolute score changes for MRR, MAP, and P@3. 43

6.4 Summary of abstracts: early precision (P@3) and depth (MRR, MAP) for classical baselines (unconstrained vs. constrained), the best single LLM (`Qwen 2.5`, Cosine–Jaccard Average), and the agentic pipeline (Cosine–Jaccard Average). Constraints amplify cosine–mean depth; dot–max remains the strongest classical top-k baseline; the agent improves early precision while cosine–mean retains the best MAP under constraints. 46

6.5 Effect of assignment constraints on classical baselines using full abstracts (constrained − unconstrained). Score changes are shown across MRR, MAP, and P@3 for each similarity and aggregation method. 48

6.6 Effect of edge structure in abstract-based knowledge graphs: absolute performance changes when collapsing typed relations to `related_to`. CSO-Classifier shows strong gains with typed edges; Open IE sees marginal P@3 gains but drops in depth metrics; GLiNER benefits on Jaccard P@3 under collapsed edges. 51

6.7 Abstract-based reviewer matching. Early precision (P@3) and depth (MRR, MAP) across classical baselines (unconstrained vs. constrained), the best KG setting (CSO classifier with relation edge; Cosine–Jaccard Average), and the agentic pipeline (Cosine–Jaccard Average). Constraints elevate Jaccard–Mean on depth and shortlist quality; Cosine–Max excels for earliest hits; CSO classifier with relation edge KGs yield the strongest overall quality; the agent is competitive on early precision. 54
6.8 Lorenz curves comparing reviewer assignment inequality across methods. 55

List of Tables

6.1 Reviewer matching using titles only and without the employment of constraints. . . . 36
6.2 Reviewer matching using titles only and with the employment of constraints. . . . 37
6.3 Reviewer matching based on LLM-derived keyword representations extracted from paper titles. . . . 39
6.4 Reviewer matching based on the LLM agentic pipeline applied to paper titles. . . . 40
6.5 Reviewer matching using the summary of the abstracts and without the employment of constraints. . . . 42
6.6 Reviewer matching using the summary of abstracts and with the employment of constraints. . . . 42
6.7 Reviewer matching based on LLM-derived keyword representations extracted from summary of abstracts. . . . 44
6.8 Reviewer matching based on the LLM agentic pipeline applied to summary of abstracts. . . . 45
6.9 Reviewer matching using full abstracts only and without the employment of constraints. . . . 47
6.10 Reviewer matching using full abstracts only and with the employment of constraints. . . . 47
6.11 Reviewer matching using KGs and relation edge. . . . 49
6.12 Reviewer matching using KGs and `related_to` edge. . . . 50
6.13 Reviewer matching based on LLM-derived keyword representations extracted from Abstracts of papers. . . . 52
6.14 Reviewer matching based on the LLM agentic pipeline applied to abstracts of papers. . . . 53

Chapter 1

Introduction

1.1 Motivation and Open Issues

High-quality peer review is the backbone of scholarly communication, shaping what knowledge is disseminated and how fast it reaches the community. Conferences and journals in fast-moving areas such as computer science, AI, and related interdisciplinary fields now receive thousands of submissions per cycle, intensifying the reviewer assignment problem (RAP), matching manuscripts to suitably qualified reviewers while respecting constraints on expertise, capacity, conflicts, and fairness. Manual assignment struggles with scale and emerging subfields; empirical evidence also shows that reviewer composition can materially affect outcomes [8, 40].

Historically, automated RAP pipelines have modeled paper–reviewer affinity with text similarity over titles and abstracts, topic models, or vector-space representations (e.g., Toronto Paper Matching System(TPMS)), then solved a constrained matching problem. These approaches brought scalability and some interpretability but face challenges as research becomes more dynamic and interdisciplinary [13, 7]. More recent contextual encoders (e.g., BERT/SciBERT) improve semantic fidelity but can reduce interpretability and auditability in practical workflows [18, 6]. Persistent pain points include: (i) incomplete or uneven reviewer evidence (often limited to titles/abstracts), (ii) noise from unrelated publications that dilute affinity, (iii) semantic drift and synonymy across subfields, (iv) auditability–performance trade-offs with black-box embeddings, and (v) operational constraints and fairness requirements in large venues [3, 90].

Motivated by these gaps, this thesis investigates content-based reviewer assignment from complementary angles that prioritize data realism, semantic fidelity, and interpretability. The work advances affinity construction using classic IR features, knowledge-graph (KG) signals, and Large Language Models (LLMs) individually and studies how these affinities feed into assignment under real-world constraints, while aligning with community guidance on limited, assistive use of generative tools in peer-review operations [42, 86].

1.2 Research Questions

To address the challenges of reviewer assignment in peer review systems, this thesis investigates key aspects of representing, estimating, and leveraging reviewer–paper affinities. The work is guided by the following four research questions:

- **RQ1 — Representation:** How do different textual representations (titles, abstracts, LLM-generated summaries) and similarity measures (e.g., cosine, dot product, Jaccard) shape paper–reviewer affinity, especially when reviewer evidence is sparse or noisy?
- **RQ2 — Robustness to incomplete/noisy profiles:** What strategies mitigate the twin problems of (i) incomplete reviewer data and (ii) interference from non-manuscript-related publications when estimating expertise?
- **RQ3 — Interpretability vs. accuracy:** Can keyword-driven or KG-based profiles, built with controllable extraction methods (OpenIE, topic classifiers, named-entity recognizers) and/or LLMs, deliver competitive assignment quality and human-readable justifications?
- **RQ4 — From affinities to assignments:** Given improved affinities, how do fairness-aware objectives and standard matching algorithms trade off coverage, equity, and reviewer workload in practice?

1.3 Methodological Overview

In order to explore the core challenges of reviewer–paper matching and to rigorously investigate the proposed research questions, this thesis designs and evaluates a set of diverse methodological pipelines. These pipelines are intended to balance interpretability, robustness, and assignment quality across a range of realistic scenarios, including sparse reviewer data and noisy profiles. By integrating traditional similarity-based methods with recent advances in large language models (LLMs), structured knowledge extraction, and fairness-aware optimization, the aim is to provide a comprehensive view of the design space for affinity modeling and reviewer assignment. The five primary methodological components are as follows:

1. **Classical content-based baselines:** Papers and reviewer profiles are represented via titles, abstracts, or LLM-derived summaries; affinities use Jaccard, dot product, and cosine similarity over TF–IDF-like features.

2. **LLM-keyword pipelines.** Multiple LLMs generate concise, normalized keyword sets for manuscripts and reviewers; affinity combines cosine similarity (over TF–IDF of keywords) and Jaccard overlap for transparent, interpretable matches.

3. **Knowledge-graph (KG) signals:** From titles/abstracts/LLM summaries, OpenIE triples are enriched with GLiNER entities and CSO topics to build structured profiles for submissions and reviewers.

4. **Agent-assisted affinity building:** A lightweight LLM agent normalizes terminology, extracts salient keyphrases, and produces brief, auditable rationales to assist human judgment.

5. **Assignment under constraints:** Affinities feed standard matching solvers with capacity and conflict constraints, with optional fairness-aware objectives that protect the worst-served papers and balance loads.

Across all pipelines, ranking quality is assessed with MRR, MAP, and Precision@k, enabling comparisons with classical and neural baselines [44].

1.4 Data and Evaluation Protocol

In this study we use a dataset in the Semantic Web and Computer Science domains comprising 663 submissions from 85 conferences and anonymized profiles for 524 reviewers. Reviewer evidence is drawn from titles and abstracts of their most-cited publications (up to 20 per reviewer), foregrounding the challenges of incomplete profiles and topic interference [3].

Affinity models are compared under controlled representation variants (title-only, abstract-only, LLM summary) and similarity measures (cosine, Jaccard, and their average). For LLM-based pipelines, multiple models are evaluated to quantify variance; for KG-based pipelines, ablations isolate contributions of OpenIE triples, CSO topics, and GLiNER entities. Where applicable, assignment-level analyses study interactions between improved affinities and fairness/load-balancing objectives [61, 84].

1.5 Contributions

Collectively, this work advances the reviewer–assignment problem along the full pipeline—from diagnosing where affinity modeling fails to demonstrating how improved signals translate into fairer, higher-quality matches. We begin by characterizing RAP under realistic constraints (sparse, noisy reviewer evidence) and by releasing resources that enable rigorous, reproducible evaluation. Building on this

foundation, we introduce interpretable affinity models that leverage LLM-derived keywords and structured knowledge to make the drivers of each match auditable. We further explore lightweight agent support to standardize terminology and surface concise rationales that assist human decision-making without exposing confidential content. Finally, we connect these signals to assignment objectives that explicitly account for equity and workload balance, revealing practical trade-offs between topical coverage and fairness. Specifically, the thesis contributes:

1. **Problem characterization under realistic constraints.** A systematic analysis of RAP when reviewer evidence is limited to titles/abstracts, highlighting failure modes due to data sparsity and interference and quantifying how representation choices affect affinity ranking.

2. **Public, domain-curated dataset and evaluation recipes.** A consolidated dataset for reviewer assignment in Semantic Web and Computer Science, together with standardized evaluation protocols (MRR, MAP, P@k) to enable reproducible comparisons of affinity models and assignment solvers.

3. **Interpretable LLM-keyword framework.** A multi-LLM pipeline that generates normalized keyword profiles for papers and reviewers and combines cosine and Jaccard similarities. The method delivers competitive ranking quality while preserving traceability (what concepts drove the match).

4. **Knowledge-graph representations for expertise.** A KG-construction track that integrates OpenIE, GLiNER, and the CSO Classifier to model concept relationships explicitly, providing structured features that complement text similarity and support richer, auditable rationales.

5. **Agent-assisted affinity building.** An LLM agent that standardizes terminology and extracts keyphrases and brief rationales, improving the usability of candidate lists without entrusting generative models with evaluative or confidential content.

6. **From signals to assignments with fairness.** An end-to-end study that feeds improved affinities into classical and fairness-aware assignment objectives, surfacing trade-offs among topical coverage, equity for the worst-served papers, and reviewer workload.

1.6 Ethical Scope and Roadmap

This thesis focuses on refining affinity estimation and assignment support in peer-review workflows, rather than replacing human judgment or automating the act of reviewing itself. Large Language Models (LLMs) are employed strictly in assistive roles such as keyword extraction, terminology normalization, and the generation of

concise, human-readable rationales to support transparency and informed decision-making by program chairs and editors. Throughout the thesis, particular attention is paid to auditability, traceability of signals, and the responsible handling of sensitive peer-review data.

Beyond model-level considerations, this work emphasizes *systemic risks inherent to peer-review processes*. These include biases in reviewer selection, uneven workload distribution, reinforcement of established research networks, underrepresentation of emerging or interdisciplinary topics, and over-reliance on opaque automated recommendations. Such risks can affect fairness, diversity, and the perceived legitimacy of the review process, even when no generative content is produced. The methods proposed in this thesis explicitly address these concerns by prioritizing interpretable representations, hybrid similarity signals, and fairness-aware assignment objectives that balance topical relevance with reviewer workload and coverage.

While technical risks associated with generative models such as hallucinations or prompt injection are acknowledged, they play a secondary role in this context, as LLMs are not used to generate evaluative judgments or review content. Instead, the ethical focus is aligned with current community policies that allow limited, assistive use of AI in peer review while preserving human oversight and accountability [24, 42]. Within this scope and ethical posture, the dissertation proceeds as follows:

- **Chapter 1** motivates the reviewer assignment problem, frames the challenges, and outlines the research questions and contributions.
- **Chapter 2** surveys related work on reviewer assignment from information retrieval, recommender systems, and operations research, situating classical approaches alongside knowledge-graph–based and LLM-assisted methods.
- **Chapter 3** studies classical content-based baselines under realistic reviewer evidence (titles, abstracts, and LLM-generated summaries), analyzing representation choices and similarity measures.
- **Chapter 4** introduces the LLM-based keyword framework, compares multiple models, and evaluates interpretability–performance trade-offs.
- **Chapter 5** develops the knowledge-graph–based track using OpenIE, GLiNER, and the CSO Classifier to build structured expertise representations and compares them with keyword-driven approaches.
- **Chapter 6** integrates agent-assisted affinity construction and investigates reviewer assignment under capacity constraints, conflict-of-interest rules, and fairness objectives, including workload balancing and protections for poorly served submissions.
- **Chapter 7** synthesizes findings into practical guidance, discusses limitations (e.g., domain transfer, multilingual settings, reviewer fatigue), and outlines

future directions such as cross-venue calibration, multilingual knowledge enrichment, and uncertainty-aware assignment.

Taken together, the thesis demonstrates that carefully engineered content signals classical, knowledge-graph–based, and LLM-assisted can substantially improve reviewer assignment while mitigating systemic peer-review risks. By foregrounding interpretability, fairness, and human oversight alongside accuracy, this work contributes practical and trustworthy building blocks for reviewer assignment in real-world scholarly venues.

Chapter 2

Related Work

The Reviewer Assignment Problem (RAP) has long posed a foundational challenge in academic peer review, requiring the selection of appropriate reviewers for submitted manuscripts under constraints of topical relevance, fairness, and integrity. Traditional approaches relied heavily on manual reviewer selection or simple keyword matching, which proved inadequate in capturing the semantic depth of scholarly content and scaling to the growing volume and complexity of modern submissions [49]. A key advance came with the development of early automated systems such as the Toronto Paper Matching System (TPMS), which utilized language model predictors and archive harvesting to improve paper–reviewer similarity scoring, laying the groundwork for further developments in semantic representation and optimization-based assignment [13].

Subsequent research emphasized semantic methods for enhancing paper–reviewer affinity modeling. Latent topic models such as Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI) were initially adopted to uncover hidden thematic structures in text. Extensions such as time-aware modeling aimed to account for temporal shifts in expertise [54, 26]. Nevertheless, these methods struggled to accommodate interdisciplinary content and reviewer ambiguity [1]. More recently, vector-based representations—such as those produced by SPECTER [14] and other deep encoders—have enabled richer semantic embeddings for both manuscripts and reviewer profiles [89].

A parallel line of work addresses the challenges of *cold-start scenarios* and *topic drift*, recognizing that both submission domains and reviewer expertise evolve over time [56]. Time-sensitive encoders and incremental profiling methods have been proposed to capture the emergence of new research areas, sabbaticals, or methodological shifts [11]. Techniques include decay-weighted profiles that emphasize recent outputs and streaming representations that dynamically update a reviewer's expertise with new publications—particularly relevant in interdisciplinary settings where scholarly trajectories often cross traditional boundaries [77].

Another area of expansion involves *multilingual* and *multimodal* reviewer matching. Cross-lingual encoders and translation-augmented pipelines support align-

ment when the language of the submission diverges from the available reviewer evidence [25]. Simultaneously, leveraging non-textual modalities—such as figures, tables, code, and datasets—has been explored to capture methodological nuances and task-specific information that may be poorly conveyed through text alone [80].

Knowledge Graphs (KGs) have emerged as powerful tools for encoding structured reviewer and author information. These graphs are particularly valuable for modeling hierarchical topics and detecting conflicts of interest [59]. Hybrid approaches that combine KGs with rule-based matching have demonstrated efficacy in aligning reviewer expertise with manuscript content, especially in interdisciplinary contexts [82]. More recent models incorporate graph-based representations of co-authorship networks, institutional affiliations, and social ties to improve robustness and fairness in reviewer selection [41].

The advent of deep learning and Large Language Models (LLMs) has substantially advanced the field. Neural models employing architectures such as CNNs, BiLSTMs, and BERT have been used to learn semantic similarity between manuscripts and reviewer publications [20]. Some approaches treat reviewer assignment as a classification task, leveraging sentence-level modeling of abstracts and titles [75]. Others have introduced hybrid frameworks that combine deep embeddings with graph-based signals [85], while LLMs have also been applied for data augmentation in low-resource domains [10]. Compared to traditional topic models, these methods demonstrate superior contextual understanding and generalization capabilities.

Keyword extraction remains a cornerstone of reviewer modeling. Classical algorithms such as RAKE and TextRank [60, 47] offer interpretable keyword lists, while more recent neural approaches adopt sequence-to-sequence architectures for keyphrase generation [46]. LLM-enhanced pipelines have further improved keyphrase extraction through prompt engineering and abstractive summarization [45]. This transition toward interpretable, model-driven keyword representations supports broader efforts to enhance transparency and trust in automated reviewer assignment systems [57].

Simultaneously, *interpretability* and *auditability* have become central concerns in the design of reviewer recommendation systems. Despite their predictive power, opaque neural similarity models raise concerns about transparency and accountability, especially when assignments appear arbitrary. As a result, recent approaches increasingly emphasize explainable representations, including human-readable keyword profiles, provenance traces of assignment decisions, and post-hoc rationales derived from model attention or prompt chains [88]. These features help area chairs and editors understand system behavior and facilitate external audits and reproducibility [48].

Beyond semantic similarity, *fairness* and *conflict mitigation* remain critical dimensions of RAP. Algorithms like PeerReview4All introduce provable guarantees for max-min fairness under both load and similarity constraints [73]. Other studies address robustness to adversarial perturbations, such as manipulated abstracts [21], or incorporate collusion-aware models that account for social and institutional reviewer

networks [31]. Conflict detection via graph-based analyses has also gained traction, enabling identification of potential biases from past collaborations or affiliations [50].

Recent research has also revisited *preference elicitation* and bidding as signals that complement content-based similarity. Techniques include reliability-weighted integration of bids, learning-to-rank models informed by historical assignments, and calibration schemes that account for overly generous or self-interested bidding behaviors [27]. These approaches treat expressed preferences as *noisy labels*, requiring careful modeling to avoid distortions [35].

Uncertainty-aware affinity modeling has likewise gained attention. Rather than relying solely on point estimates, some methods generate confidence intervals or distributional scores and incorporate these into the assignment solver [15]. Strategies such as backup reviewer allocation, risk-hedging across expertise types (e.g., methods vs. application), and risk budgeting aim to reduce over-exposure to uncertain matches while promoting fairness across submissions [16].

Ensemble-based systems represent another strand of research, where multiple reviewer attributes expertise, diversity, authority are jointly optimized [43]. For example, RevASIDE selects reviewers based on complementarity rather than similarity alone [39]. Optimization techniques, including Integer Linear Programming and Max-Flow formulations, ensure both scalability and constraint satisfaction [73]. From a systems perspective, demands for *scalability and responsiveness* have spurred research into approximate nearest neighbor (ANN) search over embeddings, cache-aware re-ranking, and incremental indexing for rolling submission cycles. These efforts decouple candidate generation from constrained optimization, supporting determinism and traceability in large-scale deployments.

On the *human-in-the-loop* front, interface-level innovations have sought to make assignment decisions more transparent and editable. Features such as compact rationales (e.g., keyphrases or representative publications), snippet-based previews, and interactive controls (e.g., pin/unpin reviewers, adjust similarity weights) empower chairs to steer the system without retraining underlying models. LLMs are increasingly embedded within these workflows as assistants that generate justifications, harmonize terminology, or draft rationales—while final authority remains with the human user [4].

Beyond the assignment task itself, LLMs have also been explored for *review generation*, *critique simulation*, and *meta-review support*. Initiatives such as ReviewAgents demonstrate that structured reviewer feedback can be emulated using prompt-tuned LLMs [28]. Other studies examine how human reviewers may benefit from AI-generated suggestions, although limitations remain in assessing originality and complex argumentation [76, 91]. These findings reinforce the role of LLMs as augmentation tools rather than full substitutes for expert judgment.

An emerging area of interest is the use of *agentic workflows*, where multiple autonomous AI agents collaborate to complete the reviewer assignment pipeline. These agents often assume specialized roles—such as keyword extraction, evidence verification, and conflict adjudication—working in concert to produce higher-quality

and more reliable recommendations. For example, adjudication agents can resolve inconsistencies among LLM-generated keywords or adjust reviewer–topic alignments using contextual feedback. These modular architectures offer a scalable path toward adaptive, transparent assignment systems with human-in-the-loop configurations that preserve editorial control [79].

Simultaneously, increasing attention is being paid to issues of *governance, privacy, and robustness*. Contributions in this space include de-identification protocols for input data, stricter conflict detection via co-authorship and institutional graphs, and defenses against adversarial tactics such as abstract polishing. Policy-aware pipelines now encode venue-specific constraints (e.g., geographical or industrial representation), while logging mechanisms support transparency and appeals. Reproducibility best practices, such as public artifact releases, leakage-aware evaluation splits, and sensitivity analyses have become critical for credible and trustworthy system development [55].

System-level surveys underscore the multidimensional nature of RAP, bringing together advances across representation learning, optimization, fairness, and evaluation [3]. Shared benchmarks, including SciRepEval and RAP-specific gold standards, provide essential infrastructure for experimental rigor [74, 69]. Evaluation metrics such as MAP, MRR, and Precision@k continue to serve as standard tools for assessing retrieval-based assignment systems [9].

Despite substantial progress, several open challenges persist. These include adapting to dynamic and interdisciplinary expertise, reducing systemic biases in reviewer modeling, and increasing transparency in opaque similarity measures. Promising directions include hybrid frameworks that integrate symbolic reasoning (e.g., rule-based engines and knowledge graphs) with neural embeddings, as well as interpretable interfaces based on human-readable keywords [19]. The growing role of LLMs in summarization, literature synthesis, and knowledge extraction also suggests a broader trend toward leveraging artificial intelligence across the entire peer review lifecycle [2, 32].

In light of these developments, our work adopts LLM-generated keyword representations as a practical and interpretable alternative to static topic models and opaque embedding-based similarity methods. Rather than relying solely on traditional content features or dense vector representations, we integrate these keywords within a broader framework that spans four complementary pipelines: classical content-based baselines, knowledge graph–enriched semantic matching, LLM-derived keyword modeling, and agentic workflows that orchestrate multiple LLMs with adjudication to generate auditable, high-quality keyword sets. By combining these representation strategies with fairness-aware assignment and hybrid scoring mechanisms, our framework offers a transparent, robust, and scalable solution to the reviewer assignment problem under realistic academic constraints.

Chapter 3

Problem Definition and Task Formulation

3.1 Reviewer Assignment Problem

The central task addressed throughout this PhD research is the automated assignment of academic reviewers to submitted research manuscripts, with an emphasis on semantic alignment, expertise relevance, and system transparency. Reviewer assignment is a critical component of the peer-review process in scholarly publishing, where the quality of evaluation hinges upon matching each submission with reviewers who possess the appropriate domain knowledge, methodological expertise, and research focus.

Formally, the task involves assigning each submitted paper $P \in \mathcal{P}$ to a fixed number of reviewers $R \subset \mathcal{R}$, where the assignment must optimize semantic relevance while satisfying fairness and integrity constraints. This overarching objective has been explored through multiple complementary methodologies developed across the course of this project.

At a high level, the targeted task comprises three core goals:

1. **Semantic Relevance**: Ensuring that the assigned reviewers are topically aligned with the submission, based on deep semantic representations of the content and reviewers' publication histories.

2. **Fairness and Constraints Compliance**: Enforcing conference-specific policies, such as conflict-of-interest avoidance and workload balancing.

3. **Interpretability and Scalability**: Developing systems that are not only accurate and robust but also interpretable and scalable for real-world academic settings.

To accomplish this, we have proposed and evaluated multiple frameworks that frame reviewer matching as a content-based recommendation problem, structured through the following subtasks.

3.2 Core Objectives of the Task

The automated assignment of reviewers is inherently a multi-objective task that balances topical relevance, procedural fairness, and system-level usability. To guide the design of all proposed solutions, this research identifies three core objectives: semantic relevance, fairness and constraint compliance, and interpretability with scalability. These objectives serve as foundational pillars for all subsequent modeling choices and evaluation criteria. Semantic relevance ensures that the expertise of assigned reviewers aligns closely with the content of submitted manuscripts, enabling high-quality peer feedback. Fairness and constraint compliance address practical requirements such as workload balancing and conflict-of-interest exclusion, both of which are critical to maintaining the integrity of the peer-review process. Lastly, interpretability and scalability prioritize the development of models that are not only effective but also transparent and feasible to deploy at scale in real-world conference or journal settings. Each component of the proposed reviewer assignment framework is developed with these three goals in mind.

3.2.1 Semantic Feature Extraction

Across all approaches, the first step involves extracting meaningful semantic features from both manuscripts and reviewer profiles. Early approaches leveraged title, abstract and summary of abstracts information combined with structured constraints, while later systems employed LLMs to generate interpretable keyword-based representations or construct knowledge graphs from scientific texts. Feature extraction pipelines included:

- LLM-generated keywords via prompt-based querying.
- Semantic triple extraction and topic enrichment.
- Domain-specific summarization using fine-tuned models (e.g., T5 on SciTLDR).
- Graph-based representations of reviewer and paper content.

3.2.2 Similarity Computation and Reviewer Matching

As a foundational approach, this research project began with a traditional content-based reviewer assignment method, where similarity between papers and reviewers was computed using vector representations (e.g., TF-IDF) and basic dot product similarity. This method established a baseline for later, more advanced models, allowing for clear comparisons in performance, interpretability, and semantic depth.

To determine the relevance between a submission and potential reviewers, we designed multiple similarity computation strategies:

- **Vector-space models** using TF-IDF representations with **cosine similarity**.
- **Set-based models** using **Jaccard similarity** over extracted keywords.
- **Dot product similarity** as a simple yet effective baseline measure in early-stage experiments.
- **Hybrid similarity measures** combining lexical and semantic comparisons.
- **Node embeddings** derived from knowledge graphs to capture latent structure.

These methods support ranking all candidate reviewers for each paper, forming the basis for selecting the most topically aligned experts.

3.2.3 Assignment under Constraints

To ensure fairness, integrity, and adherence to academic review standards, the assignment task incorporated the following constraints:

- Each paper must be assigned to exactly **three reviewers**.
- Each reviewer may review a **limited number of papers** (e.g., up to three).
- Reviewers with a **prior collaboration** history with the paper's authors are **excluded** to avoid conflicts of interest.

These constraints were enforced either directly during the assignment phase or indirectly through pre-filtering and matching logic.

3.3 Evaluation and Ranking

All proposed systems were rigorously evaluated using information retrieval (IR) metrics to assess the quality of reviewer matches:

- **Mean Reciprocal Rank (MRR)** for top-ranked reviewer relevance.
- **Precision@K (P@K)** to evaluate the quality of top recommendations.
- **Mean Average Precision (MAP)** for holistic ranking performance.

Evaluations were conducted against historical reviewer assignment datasets.

This work positions the reviewer assignment problem as a semantically grounded, constraint-aware, and data-driven matching task, progressively enhanced through the integration of advanced NLP techniques and large language models. Each system developed within this research contributes to a modular, interpretable, and scalable pipeline capable of improving reviewer recommendation quality in modern academic publishing workflows.

Chapter 4

Reviewer Assignment Pipeline

4.1 Pipeline Overview

The reviewer–assignment pipeline developed in this dissertation converts raw submissions and reviewer corpora into ranked, *auditable* recommendations. Its design is guided by three principles: *interpretability*—exposing explicit features and per–match rationales for every recommendation; *portability*—minimizing domain–specific tuning via modular preprocessing and model–agnostic similarity functions; and *rigor*—producing reproducible artifacts and adhering to standard evaluation protocols. Together, these principles yield a transparent, transferable, and methodologically sound system suitable for deployment across venues and datasets.

As indicated in figure 4.1 the consolidated pipeline is modular and proceeds through the following blocks: (i) data resources and construction; (ii) text normalization and preprocessing; (iii) representation building via three complementary tracks content-based vectors, knowledge graphs (KGs), and interpretable LLM-induced keywords; (iv) similarity computation and score fusion; and (v) ranking and constrained assignment.

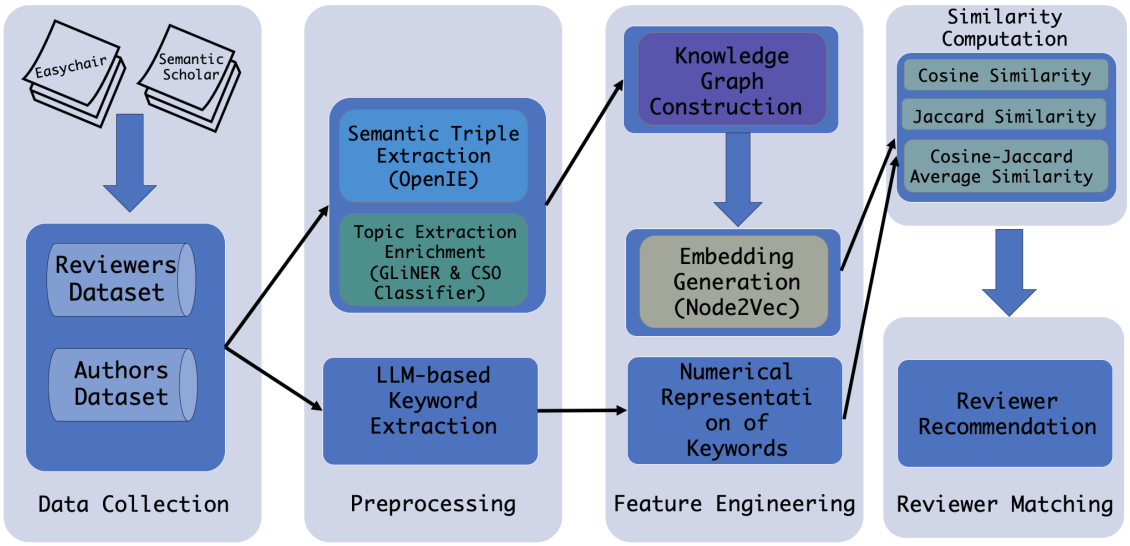

Figure 4.1: Reviewer Assignment Pipeline.

All steps are instrumented to serialize intermediate outputs, such as normalized text, keyword sets, graph features, and similarity matrices—allowing them to be inspected or replaced without impacting other components.

4.2 Data Collection

We build two interconnected resources: a corpus of submitted papers (*Authors' Dataset*) and a corpus of reviewer profiles (*Reviewers' Dataset*). The data collection process was carried out by the PhD candidate and builds upon datasets initially created in the context of prior research reports and peer-reviewed publications related to this thesis.

Process and permissions. Submission data were obtained from the EasyChair platform[1] for a set of conferences in which the supervisors of this thesis were involved as organizers, program committee members, or chairs. Access to these conferences and their submission data was therefore granted through legitimate organizational roles or explicit authorization from the respective program chairs. To facilitate systematic extraction and structuring of the data, the Selenium WebDriver tool was used to automate the retrieval of information from the EasyChair web interface within the scope of the authorized access. No data outside these permissions were collected, and no circumvention of access controls was performed.

Using this process, we collected a total of **663** submitted papers. For each paper, we recorded the conference name, paper title, link, list of authors, abstract, and the

[1] `https://www.easychair.org/`

reviewers assigned by the conference management system. All collected records were subsequently anonymized, serialized into JSON format, and linked through stable identifiers to support end-to-end traceability across preprocessing, matching, and evaluation. Data handling followed applicable privacy regulations and ethical guidelines governing peer-review data.

4.2.1 Authors' Dataset

The Authors' Dataset contains titles and abstracts for 663 submissions drawn from **85** computer-science conferences spanning Semantic Web and Linked Data, Artificial Intelligence and Machine Learning, Databases and Information Systems, Computational Linguistics and NLP, and Ontologies and Knowledge Graphs. Each entry is stored under a persistent dictionary key and includes structured metadata such as the conference name, paper title and link, authors, abstract, and the list of assigned reviewers as recorded in EasyChair. This dataset provides both the textual material used for modeling paper topics and the ground truth assignments used for evaluation.

4.2.2 Reviewers' Dataset

Starting from the unique set of reviewer names extracted from the EasyChair assignments (**524** distinct reviewers), we constructed individual reviewer profiles using the Semantic Scholar platform[2] and its public API[3]. For each reviewer, we retrieved up to 20 publications, prioritizing the most-cited works. For each publication, we stored the title, abstract, venue or link, total citation count, and number of highly influential citations. Reviewer profiles are indexed internally using stable identifiers derived from Semantic Scholar author records, while only anonymized identifiers are retained in the research dataset.

Although each paper was nominally assigned three reviewers, the final dataset comprises **524** unique reviewer profiles rather than the 1,989 implied reviewer–paper pairs. This reduction results from overlapping reviewer assignments across papers, name disambiguation issues, incomplete publication metadata, and fragmented or duplicate author entries in open bibliographic sources. These factors reflect common limitations of large-scale scholarly data and are explicitly accounted for in the experimental analysis.

4.2.3 Integration, Format, and Availability

Both datasets are stored in JSON format with stable keys to ensure reproducible linkage across preprocessing, matching, and evaluation pipelines. Paper and reviewer

[2] `https://www.semanticscholar.org/`
[3] `https://api.semanticscholar.org/api-docs/`

records are cross-referenced through anonymized EasyChair assignment identifiers, and consistent mappings are maintained to support error analysis and ablation studies.

Due to confidentiality constraints inherent to peer-review management systems and agreements with conference organizing committees, the raw submission and reviewer-assignment data cannot be publicly released. However, anonymized and aggregated versions of the datasets, together with preprocessing scripts, can be made available by the authors upon reasonable request for non-commercial academic research. The disciplinary scope of the dataset (computer science) and its sampling should be considered when interpreting the generalizability of fairness-related results.

4.3 Preprocessing

Prior to constructing knowledge graphs and executing the reviewer assignment process, a comprehensive preprocessing phase is conducted to transform unstructured textual inputs into structured, semantically enriched representations. This phase is essential for extracting meaningful concepts, identifying latent relationships, and preparing the data for similarity-based analysis.

The preprocessing pipeline consists of five key components:

1. **Semantic Triple Extraction:** Raw abstracts are first processed using OpenIE to extract fundamental (subject, relation, object) triples, forming the initial structural representation of textual content.

2. **Graph-Based Named Entity Recognition (GLiNER):** The extracted subjects and objects are further enriched using GLiNER, which integrates linguistic features and domain knowledge to identify fine-grained scientific entities such as methods, tasks, and tools.

3. **Topic Classification via CSO:** To ensure ontological consistency and enable structured comparisons across documents, topics identified in the triples are mapped to the Computer Science Ontology (CSO) using the CSO Classifier.

4. **LLM-Based Keyword Extraction:** A suite of large language models including GPT-4o-mini, LLaMA 3.2, Qwen 2.5, Phi 3.5, and Gemma 2 are used to generate concise keyword sets from paper abstracts and reviewer publications, providing interpretable and context-aware semantic descriptors.

5. **LLM Agentic Pipeline:** A multi-model pipeline built with LangChain and chat APIs generates keyword profiles by prompting several LLMs in parallel and selecting the best output using a judge model. This approach ensures concise, high-quality, and machine-readable keywords for both papers and reviewers.

Collectively, these preprocessing components yield a unified, semantically grounded representation of both papers and reviewer profiles. This representation serves as the foundation for downstream tasks such as knowledge graph construction, content similarity measurement, and ultimately, expert reviewer assignment.

4.3.1 Semantic Triple Extraction (OpenIE)

To extract structured semantic information from unstructured textual content, this study employs the Open Information Extraction (OpenIE) framework[4], which identifies relational tuples from natural language text in the form of *(subject, relation, object)* triples. OpenIE operates in an unsupervised manner, leveraging syntactic parsing and semantic role labeling to capture propositions, factual statements, and entity relationships embedded within sentences [22, 23].

Applied to the abstracts of both papers and reviewer publications, OpenIE generates a set of semantic triples that serve as an initial structured representation of the core informational content. These triples form the foundational layer for downstream knowledge graph construction by transforming free-form text into analyzable semantic structures.

However, despite its utility, OpenIE outputs are known to exhibit limitations, particularly when applied to complex scientific text. Common issues include boundary errors, generic or inconsistent relation labels, and unresolved coreference. To mitigate the impact of these issues on subsequent stages of the pipeline, several post-processing steps are applied:

- Malformed spans and weakly supported relation edges are pruned to reduce semantic noise.
- Heterogeneous or sparse relations may be collapsed into a generic `related_to` relation to retain structural coherence.
- Node labels are normalized and enriched using domain-specific taxonomies and tools such as CSO and GLiNER, improving consistency across concepts.

This semantic triple extraction step thus provides a candidate structured view of the text, forming the basis for graph-based modeling of content. While inherently noisy, its utility is enhanced through careful post-processing and integration with robust similarity measures (e.g., cosine, Jaccard, and their average), which reduce sensitivity to individual extraction errors and allow for meaningful reviewer-paper comparisons in later stages of the system [5].

[4] `https://nlp.stanford.edu/software/openie.html`

4.3.2 Graph-Based Linguistic Named Entity Recognition (GLiNER)

To enrich the semantic representation of extracted triples and align them with domain-specific ontologies, this work incorporates the use of Graph-based Linguistic Named Entity Recognition (GLiNER) [83]. GLiNER is a state-of-the-art named entity recognition (NER) system designed specifically for low-resource or specialized domains. It integrates transformer-based contextual encoders with external knowledge sources such as domain-specific gazetteers and knowledge bases (e.g., the Computer Science Ontology, or CSO [84]) to improve the identification and classification of fine-grained entities in scientific text.

Unlike traditional NER systems, which often struggle with domain-specific terminology, GLiNER dynamically incorporates structured background knowledge during training. It enables the recognition of nuanced entity types such as *tasks*, *methods*, *tools*, and *datasets*, thereby offering superior precision and recall in scientific domains. Evaluations across multiple fields, including computer science, biomedicine, and the social sciences demonstrate GLiNER's robustness and versatility [83, 84, 36, 81]. Given its ability to link entities to structured ontological concepts, GLiNER has become increasingly relevant for downstream applications such as knowledge graph construction, information extraction, and expert profiling.

In the present system, GLiNER is applied to the **subjects and objects** of semantic triples produced by OpenIE (Section 4.3.1). For each term, GLiNER assigns one or more domain-relevant labels along with a confidence score $s \in [0, 1]$, representing the model's posterior belief in the correctness of each classification. Higher values of s indicate stronger confidence in the entity–label assignment. This semantic annotation process enhances the informativeness of the triples and facilitates alignment with existing ontology nodes where applicable.

Implementation Details. Two key parameters were configured for GLiNER's integration into the pipeline:

- **Threshold Setting:** A confidence threshold of 0.1 was selected to control the sensitivity of entity detection. This relatively low threshold favored recall over precision, ensuring that a wide range of potentially relevant concepts were captured. While this increases the likelihood of false positives, it supports a more comprehensive representation of scientific content in downstream matching tasks.

- **Domain-Specific Labels:** GLiNER was restricted to identify entities from a predefined set of categories: "tasks", "methods", "tools", and "computer science". This focused the entity recognition process on semantic features most relevant to the reviewer recommendation problem.

Example. To illustrate GLiNER's enrichment process, consider a sample triple with the subject "Such models" and the object "levels on tasks." After processing with GLiNER, the subject is labeled as a "tool," and the object as a "task." These enriched labels provide semantic context to otherwise ambiguous textual spans, transforming the raw triple into a more meaningful representation. As a result, the system gains a deeper understanding of the entities involved, improving downstream operations such as similarity computation and knowledge graph integration.

4.3.3 CSO Classifier

To semantically annotate scientific content with standardized research topics, this work incorporates the CSO Classifier, an unsupervised system designed to assign topics from the Computer Science Ontology (CSO) [64]. The CSO is a large-scale, hierarchical taxonomy that models research areas within computer science. It comprises tens of thousands of interconnected topics and is continuously expanded using automated algorithms such as Klink-2 [52], which infer relationships based on patterns in scholarly metadata, citation networks, and lexical co-occurrence.

The CSO Classifier operates in three distinct modes [64, 61]:

- **Syntactic mode**, which detects explicit string matches of CSO labels in the input text.
- **Semantic mode**, which applies distributional semantics to identify conceptually similar terms that may differ lexically.
- **Combined mode**, which integrates both strategies for enhanced coverage and precision.

This hybrid design enables the classifier to recognize both surface-level topic mentions and more implicit conceptual associations, allowing for robust topic assignment even when the terminology used in a text varies from the ontology labels.

The CSO Classifier has been successfully deployed in various large-scale applications, including scholarly knowledge graph (KG) construction [62], research trend analysis, and editorial support systems for scientific publishers [53, 63]. Its ability to map extracted entities and concepts to a shared, evolving ontology enables consistent representation of research domains, making it especially valuable in contexts requiring longitudinal analysis or cross-dataset integration.

Application in This Work. In the present system, the CSO Classifier is applied to semantic elements extracted from paper abstracts, particularly the subjects and objects within OpenIE triples (see Section 4.3.1). For each relevant phrase, the classifier assigns one or more CSO topic labels and maps them to the ontology's hierarchical structure. This process serves two main purposes:

1. It enriches the semantic representation of extracted content by linking general research areas to more specific subfields.

2. It enables integration of these topics into downstream knowledge graph construction, facilitating structured representation and similarity computation.

By anchoring concepts to an ontology like the CSO, the system benefits from greater semantic interoperability, improved interpretability, and alignment with standard research taxonomies—key advantages for reviewer assignment and other academic recommendation tasks.

4.3.4 LLM-Based Keyword Extraction

To complement the knowledge-graph-based approach, this study employs a language model-based pipeline to generate interpretable keyword representations for both papers and reviewer profiles. Specifically, we utilize a set of instruction-tuned LLMs for the task of keyword extraction, including GPT-4o-mini, LLaMA 3.2, Qwen 2.5, Phi 3.5, and Gemma 2. These models were selected to ensure diversity in architecture, training corpus, and reasoning ability, while adhering to computational constraints of local or API-based deployment.

Each model was prompted with carefully designed templates to extract contextually relevant keywords from the *title*, *abstract*, and *summary of abstracts* of scientific documents. For example:

- **For authors' papers:**

  > *"Generate a set of keywords that would help find a reviewer for the following paper* `text_type`*: '*{`text`}*'."*

- **For reviewers' publications:**

  > *"Propose a set of keywords that define the expertise of the author who wrote a paper with the following* `text_type`*: '*{`text`}*'."*

In both cases, the placeholder {`text`} was dynamically substituted with the paper's title, summary, or abstract, depending on the context. This template-based prompting was used consistently across all LLMs, enabling a fair comparison of their keyword generation capabilities. Model inference was conducted locally where possible (e.g., via the Ollama framework for LLaMA, Qwen, Phi, and Gemma), and through API-based endpoints for GPT-4o-mini.

Summarization via T5. To assist in condensing lengthy abstracts into more digestible inputs for keyword generation, we used the T5-base model[5], fine-tuned on

[5] `https://huggingface.co/docs/transformers/en/model_doc/t5`

the SciTLDR dataset[6], which provides high-quality, domain-specific summaries of scientific papers. T5 was chosen specifically for its performance on text-to-text tasks such as abstractive summarization. Summaries were constrained to a maximum of 150 tokens to ensure brevity while preserving core content. These distilled versions were used as inputs to the LLMs for downstream keyword extraction.

Postprocessing and Normalization. Due to the inherent variability of LLM outputs, a robust postprocessing pipeline was applied. Initially, raw keyword outputs were cleaned using the Natural Language Toolkit (NLTK)[7]. This included tokenization, lowercasing, punctuation removal, and whitespace normalization. Stopwords were filtered out to retain only semantically meaningful terms, followed by lemmatization and stemming to unify word forms. Additionally, spell-checking routines were used to correct typographical errors, further enhancing the quality and reliability of the extracted keywords.

Keyword Representation. Instead of using deep contextual embeddings (e.g., transformer-based sentence or token embeddings), we represent the extracted keywords using the classic TF-IDF scheme [65]. This deliberate choice ensures transparency, reproducibility, and isolation of the contribution of the keyword generation step from the representational layer. By avoiding more complex vector encodings, we focus the evaluation on the relevance and accuracy of the LLM-generated keywords themselves.

Motivation for Model Selection. The use of multiple LLMs allowed us to explore model diversity and robustness in keyword generation. LLaMA 3.2, Qwen 2.5, Phi 3.5, and Gemma 2 were primarily chosen for their open-source availability and ability to run locally, offering scalability without reliance on commercial APIs. GPT-4o-mini was included for benchmarking purposes due to its strong performance in language understanding. While other LLMs were used extensively in early experiments due to its performance efficiency, trade-off and ease of deployment, the broader inclusion of alternative models enables cross-model comparison and supports future modularity in pipeline design.

Role in Reviewer Matching. The LLM-generated keywords serve as interpretable, semantically rich descriptors of both paper content and reviewer expertise. These keyword sets form the input for similarity computation (see Section 4.3.7), where they are used to quantify the topical alignment between submissions and potential reviewers. By providing high-level yet content-driven representations, this approach enhances both the efficiency and precision of the reviewer assignment process, particularly in large-scale, domain-specific academic settings.

[6] `https://huggingface.co/datasets/allenai/scitldr?row=0`

[7] `https://www.nltk.org/`

4.3.5 LLM Agentic pipeline

We designed an LLM-based agentic pipeline to construct compact, information-rich keyword profiles for both submissions and reviewers, which are subsequently matched using standard similarity measures. The pipeline integrates two scripts that coordinate a small ensemble of chat models and a dedicated "judge" model, implemented through LangChain[8] and standard chat APIs (OpenAI/OpenRouter[9]). Each submission and reviewer profile is processed separately for each *text_type* (title, abstract, and summary-of-abstract).

For every input text, the system issues the same production prompt to multiple models (e.g., Llama 4, Qwen 3, Phi 4, Gemma 3, Grok 4, Gemini 2.5 and GPT-4o-mini chat models exposed via OpenRouter/OpenAI), collects candidate keyword lists, and then asks a stronger "judge" model (GPT 5) to select the best candidate. The production prompt enforces a compact style and machine-parseable output:

> *"You are helping extract search keywords for assigning expert reviewers. Write 6–12 compact, specific keywords (1–3 words each) relevant to the paper* ***text_type****. Avoid generic words; prefer uni/bi/tri-grams. Output* ***must*** *be a JSON array of lowercase strings with no duplicates.* ***text_type:*** *{title—abstract—summary}. TEXT: '{text}'."*

The judge prompt receives generated keywords from all LLMs, returning the list of the keywords as JSON and records the selected one. We used LangChain's parallel runnables to execute producers concurrently and a single judge call to minimize latency.

4.3.6 Knowledge Graph Construction

To semantically represent the content of both scientific papers and reviewer profiles, this work constructs document-level knowledge graphs (KGs) based on enriched semantic triples. These triples are extracted through a pipeline involving OpenIE-based relation extraction and topic enrichment via GLiNER and the CSO Classifier (see Sections 4.3.1, 4.3.2, and 4.3.3). The resulting KGs encode domain-specific entities and their relationships, providing a structured, interpretable representation suitable for downstream matching and comparison tasks.

Graph Definition. Each knowledge graph is a directed, labeled graph where nodes represent topics or concepts, and edges capture the semantic relations between them. The construction follows these principles:

- **Nodes:** Each node corresponds to a topic identified in either the subject or object position of the OpenIE triples, and enriched with semantic labels

[8] `https://python.langchain.com`
[9] `https://openrouter.ai`

using GLiNER and the CSO Classifier. These nodes encapsulate fine-grained scientific concepts such as methods, tools, or research tasks.

- **Edges:** Directed edges connect the subject node to the object node of each triple. Two configurations are considered:

  1. *Relation-specific edges:* Each edge retains the original predicate extracted by OpenIE, preserving the granularity of the identified relation.
  2. *Collapsed edges:* All predicates are replaced with a generic relation label, `related_to`, resulting in a structurally consistent graph where each triple contributes a uniform semantic link.

  Although both configurations result in the same number of edges, the second offers a simplified representation that reduces noise from unreliable or overly specific relation labels.

Graph Embedding with Node2Vec. To enable quantitative comparison of KGs, each graph is embedded into a continuous vector space using the Node2Vec algorithm [29]. Node2Vec performs biased random walks over the graph structure and applies the skip-gram model to learn latent vector representations for each node. These embeddings capture both local (neighborhood-level) and global (structural) properties of the graph.

Node2Vec was configured with the following hyperparameters: embedding dimension set to 64, walk length of 10, and 50 random walks per node. These settings were chosen to balance embedding quality and computational efficiency. For each knowledge graph, the Node2Vec model is trained independently to produce vector representations for all topic nodes.

Graph-Level Representation. To derive a single embedding for each document-level graph, we aggregate the node embeddings by computing their mean. Formally, given a graph with N nodes and node embeddings $\{v_1, v_2, \ldots, v_N\}$, the graph-level embedding G is defined as:

$$G = \frac{1}{N} \sum_{i=1}^{N} v_i.$$

This averaging process yields a fixed-dimensional vector that serves as the semantic signature of the document, enabling efficient comparison between papers and reviewers based on their graph structures.

Methodological Justification. Node2Vec was selected over more complex graph neural networks (e.g., Graph Attention Networks or GraphSAGE) for several reasons. First, the task is entirely unsupervised, and the graphs constructed are relatively small and numerous (one per document). Second, Node2Vec offers a stable

and computationally efficient solution that scales well to large numbers of graphs while still capturing meaningful structural patterns. These characteristics make it well-suited for content-based reviewer assignment systems where interpretability, modularity, and efficiency are critical.

4.3.7 Similarity Computation and Reviewer Recommendation

To quantify the topical alignment between candidate papers and reviewers, we employ multiple similarity measures that capture both distributional and conceptual aspects of scientific content. These measures include cosine similarity, Jaccard similarity, combined similarity(average of cosine and jaccard similarities) and dot product.

Let V denote the shared vocabulary derived from the combined corpus of paper and reviewer texts. Each paper d and reviewer r are represented in two ways: (i) as non-negative TF–IDF vectors $x, y \in \mathbb{R}_{\geq 0}^{|V|}$ to encode distributional similarity, and (ii) as lemmatized, deduplicated keyword sets $P, R \subseteq V$ to capture explicit term overlap.

For every candidate paper d, we compute pairwise similarity scores with all reviewers r across the selected metrics, rank reviewers in descending order of similarity, and return the top-K most relevant candidates for recommendation. This flexible multi-metric framework enables robust reviewer matching under varying levels of textual granularity and abstraction.

4.3.7.1 Cosine Similarity

Cosine similarity measures the angular distance between two high-dimensional, non-negative vectors [72]:

$$S_{\cos}(x, y) = \frac{x \cdot y}{\|x\|_2 \, \|y\|_2} = \frac{\sum_{t \in V} x_t y_t}{\sqrt{\sum_{t \in V} x_t^2} \, \sqrt{\sum_{t \in V} y_t^2}}.$$

With TF–IDF representations,

$$x_t = \mathrm{tf}(t, d) \cdot \log\left(\frac{N}{1 + \mathrm{df}(t)}\right),$$

and similarly for y, where N is the number of documents used to train the vectorizer and $\mathrm{df}(t)$ denotes the number of documents containing term t. As all weights are non-negative, the cosine similarity $S_{\cos}(x, y) \in [0, 1]$.

This measure highlights rare but informative terms, a shared term with high inverse-document frequency contributes more strongly to the numerator, increasing similarity even when overall vocabulary differs [71, 58]. All vectors are implicitly L_2-normalized, and $S_{\cos}(x, y)$ is set to 0 if either vector is a zero vector (i.e., empty after preprocessing).

4.3.7.2 Jaccard Similarity

Jaccard similarity quantifies the degree of exact term overlap between two sets[34]:

$$S_{\text{Jac}}(P, R) = \frac{|P \cap R|}{|P \cup R|} \in [0, 1].$$

Here, P and R are sets of normalized n-gram keywords (typically unigrams to trigrams) derived from document titles, abstracts, or summaries. Terms are lowercased, lemmatized, stripped of punctuation, and deduplicated before scoring. This measure rewards precise matches in topic phrases (e.g., "neural architecture search", "semantic similarity") and penalizes overly broad keyword sets by enlarging the denominator [**?**].

In rare cases where both P and R are empty (i.e., contain no recognized keywords), we define $S_{\text{Jac}}(P, R) = 0$. Though weighted variants exist (e.g., accounting for frequency or semantic proximity), we use the unweighted set form for simplicity, interpretability, and reproducibility.

4.3.7.3 Combined Similarity

To combine the distributional robustness of cosine similarity with the conceptual precision of Jaccard similarity, we compute their arithmetic mean:

$$\text{Combined Similarity} = \frac{\text{Cosine Similarity} + \text{Jaccard Similarity}}{2} = \frac{S_{\cos}(x, y) + S_{\text{Jac}}(P, R)}{2}.$$

This fusion approach balances surface-level lexical overlap with deeper vector-space similarity. The result is intuitive, to score highly, a reviewer must both be close to the paper in a latent space and share explicit terms. This dual-signal model enhances robustness across noisy or sparse inputs.

4.3.7.4 Dot Product Similarity

The dot product, also referred to as the inner product, is one of the most fundamental and interpretable methods for measuring the similarity between two vectors [66, 67]. It is calculated by multiplying corresponding components of two vectors and summing the results. In vector space models for text analysis, such as TF-IDF representations of documents and reviewer profiles, the dot product quantifies the overlap of features across the two entities [51].

In the context of our reviewer assignment system, we use Dot Product Similarity as a baseline metric to assess how closely a reviewer's publication vector aligns with a paper's vector representation. The mathematical formulation is given by:

$$\text{DotSim}(\vec{p}, \vec{r}) = \vec{p} \cdot \vec{r} = \sum_{i=1}^{n} p_i \cdot r_i$$

where $\vec{p} = (p_1, p_2, \ldots, p_n)$ represents the vectorized form of a paper (e.g., TF-IDF values of its terms), and $\vec{r} = (r_1, r_2, \ldots, r_n)$ represents the vector of a reviewer's publication profile. The output is a scalar similarity score, higher values indicate stronger alignment between the topics in the paper and the reviewer's expertise.

Chapter 5

Evaluation

5.1 Overview

This chapter defines the protocol we use to assess our reviewer–recommendation system along three axes that matter in practice: (i) *ranking quality*—does the system surface appropriate reviewers early and consistently; (ii) *operational realism*—does it distribute work at scales seen in real venues; and (iii) *equity*—does it avoid concentrating assignments on a small subset of reviewers. We adopt standard ranking metrics from information retrieval (MRR, P@k, and MAP) to quantify accuracy against historical ground truth, and we complement them with workload statistics (macro/pooled averages and the Gini coefficient) to measure efficiency and fairness of the induced assignment distributions. All metrics are computed under the same data, constraints, and scoring schemes used throughout the paper, enabling transparent and reproducible comparisons.

Chapter roadmap. Section 5 first introduces the ranking metrics we report and the perspective each provides. Section 5.3 then formalizes our workload and fairness measures. Finally, the *Experiments* section applies this framework to the representations and similarity schemes described earlier, evaluating performance against EasyChair-derived ground truth with common operational constraints.

5.2 Evaluation Metrics

To rigorously evaluate the effectiveness of our reviewer assignment approach, we rely on three widely adopted ranking metrics: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision at K (P@K). As briefly mentioned in section 3.3 each of these metrics offers a different perspective on system performance:

1. **MAP:** evaluates overall ranking accuracy across all relevant reviewers.

2. **MRR** focuses on the rank of the first correct reviewer.

3. **P@K** assesses the quality of top-K suggestions in a practical operational context.

These metrics are designed to assess the ranking quality of predicted reviewer lists relative to the ground-truth reviewer assignments made by human chairs and widely used in information retrieval and recommendation systems, and are well-suited for evaluating ranked lists of candidate items [68, 30].As mentioned above ,each metric captures a different aspect of relevance and ordering, providing a nuanced view of how well the system performs in identifying appropriate reviewers.

5.2.1 Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR)[1] measures how highly the first relevant reviewer appears in the recommendation list [17]. For each manuscript, it computes the reciprocal of the rank at which the first ground-truth reviewer appears, and averages this value across all manuscripts:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Here, $|Q|$ is the number of manuscripts, and rank_i is the position of the first relevant reviewer for the i-th manuscript. An MRR value close to 1.0 indicates that relevant reviewers are consistently appearing at or near the top of the predicted list, while lower values suggest delayed or missed retrieval [78]. This metric is particularly suitable for applications where early retrieval of at least one relevant reviewer is critical. However, MRR does not consider the ranks of additional relevant reviewers beyond the first, and thus may not capture overall ranking quality.

5.2.2 Precision at k (P@k)

Precision at K (P@K) quantifies how many of the top K predicted reviewers are actually relevant. This metric provides a direct answer to the practical question: "If we were to select only the top K reviewers suggested by the system, how many would be correct?", in other words, it assesses the proportion of relevant reviewers retrieved within the top-k positions of the recommendation list:

$$\text{P@k} = \frac{1}{k} \sum_{i=1}^{k} \begin{cases} 1, & \text{if } r_i \in T \\ 0, & \text{otherwise} \end{cases}$$

In this expression, $r_i$denotes the reviewer ranked at position i, and T is the set of relevant reviewers for a given manuscript. P@k values fall between 0 and 1, where

[1] `https://www.evidentlyai.com/ranking-metrics/mean-reciprocal-rank-mrr`

high P@K scores indicate that the model is capable of prioritizing relevant reviewers early in the list, which is a desirable trait in reviewer assignment scenarios [37].

For our experiments, we set $k = 3$, reflecting the common academic practice of assigning three reviewers per paper. A high P@3 score thus indicates that the recommendation method can successfully place most of the relevant reviewers within the top three suggestions.

5.2.3 Mean Average Precision (MAP)

Mean Average Precision (MAP)[2] serves as a comprehensive measure of how well the system ranks all relevant reviewers across the entire list. It considers both the presence of relevant reviewers and their positions in the ranking. For each candidate paper, we first compute the Average Precision (AP), which calculates the average of precision scores at the positions where relevant (i.e., ground-truth assigned) reviewers appear in the predicted list [67]. The final MAP score is then obtained by averaging the AP values across all papers in the test set. This metric rewards systems that rank all relevant reviewers highly and penalizes those that place them lower in the list. It is formally defined as [33]:

$$\text{MAP@k} = \frac{1}{n} \sum_{i=1}^{n} \text{AP@k}_i$$

Where n is the total number of manuscripts, and AP@k_i is the average precision at cutoff k for the i-th manuscript [87]. Unlike MRR and P@k, MAP accounts for multiple relevant reviewers and their ranks, thus offering a more holistic measure of ranking quality.

For example, if a manuscript's relevant reviewers are ranked 1, 3, and 4, P@3 would yield a score of $\frac{2}{3} \approx 0.66$, as only two are within the top 3. However, MAP would also consider the fourth-ranked reviewer, yielding a higher score of approximately 0.80. This sensitivity to multiple relevant positions makes MAP particularly useful for understanding the cumulative relevance of ranked recommendations.

Together, MRR, P@3, and MAP offer complementary perspectives on performance: MRR captures early relevance, P@3 focuses on top-k accuracy, and MAP reflects the overall ranking quality. These metrics enable us to robustly evaluate how well each model can recommend suitable reviewers by learning semantic similarities between manuscript content and reviewer expertise.

5.3 Workload and Fairness Metrics

We assess the fairness and efficiency of reviewer assignments using complementary workload metrics. Let $w_{r,c}$ denote the number of papers assigned to reviewer r

[2] `https://www.evidentlyai.com/ranking-metrics/mean-average-precision-map`

in conference c. All quantities are computed on a per-conference basis over the active set of reviewers (those with $w_{r,c} \geq 1$) and then summarized across conferences. We report the *Per-Conference Average Workload*, the *Macro-Average Workload* (typical-conference view), the *Pooled Mean Workload* (system-wide view), and the *Gini Coefficient* to quantify inequality in the distribution of assignments.

5.3.1 Per-Conference Average Workload

For each conference c, let R_c be the set of reviewers who receive at least one assignment in that conference and $w_{r,c}$ the number of papers assigned to reviewer r. The *per-conference average workload* captures the typical load borne by active reviewers in c and is defined as [38]:

$$\bar{w}_c = \frac{1}{|R_c|} \sum_{r \in R_c} w_{r,c}$$

This statistic normalizes by the number of participating reviewers in c, providing a size-aware summary of workload within a single venue while excluding inactive reviewers (those with zero assignments).

5.3.2 Macro-Average Workload

To summarize workloads uniformly *across* venues, we compute the unweighted average of the per-conference means over the set $\mathcal{C}'$ of conferences with at least one assignment [70]:

$$\text{MacroAvg} = \frac{1}{|\mathcal{C}'|} \sum_{c \in \mathcal{C}'} \bar{w}_c$$

By assigning equal weight to each conference irrespective of size, the macro-average reflects the workload of a "typical" conference and prevents large venues from dominating the summary.

5.3.3 Pooled Mean Workload

To characterize the system-wide average number of assignments per active reviewer–conference pair, we aggregate all assignments and normalize by the total count of active reviewer–conference pairs [70]:

$$\text{Pooled} = \frac{\sum_c \sum_{r \in R_c} w_{r,c}}{\sum_c |R_c|}$$

Unlike the macro-average, this *pooled mean* weights conferences proportionally to the number of active reviewers, thereby emphasizing larger venues and yielding a population-level workload estimate.

5.3.4 Gini Coefficient (Workload Inequality)

To assess equity in the distribution of assignments, we compute the Gini coefficient G over the multiset $\{w_{r,c}\}$ of nonzero reviewer–conference workloads [12]:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}}$$

Here, x_i denotes an individual workload, $\bar{x}$ their mean, and n the number of active reviewer–conference pairs. A value of $G = 0$ indicates perfectly even workloads, while values closer to 1 indicate greater inequality.[3]

5.4 Experiments

Using the datasets in Section 4.2, we evaluate four complementary families of signals for reviewer assignment across our four studies: (i) classical lexical baselines, (ii) ontology-/KG-enriched content representations, (iii) LLM-derived keywords, and (iv) an agentic LLM pipeline for keyword induction and adjudication. The gold standard is the historical assignment extracted from EasyChair within the Authors' dataset; a predicted reviewer is considered relevant if they appear among the original assignees. Each submission is represented by its title, full abstract, and summary of abstract of papers (summaries were produced with a T5 model fine-tuned on SciTLDR[4]dataset) to provide compact yet faithful signals. For lexical baselines we compute Cosine similarity,Jaccard simialrity, and Dot Product similarity between paper and reviewer datasets; for KG variants we construct reviewer/paper graphs with OpenIE, GLiNER, and CSO (typed) relations, and we also consider a `related_to` collapse to probe structure sensitivity. For LLM-based representations, we prompt multiple recent models (GPT–4o-mini, LLaMA 3.2, Qwen 2.5, Phi 3.5, Gemma 2) to generate normalized keyword sets for papers and reviewers, then score Cosine (TF–IDF over keywords), Jaccard (set overlap), and their unweighted average ("Combined") to obtain ranked reviewer lists. These LLM-driven experiments span 663 papers from 85 conferences and emphasize interpretability of the matching signal. In the agentic setting, we further ensemble multiple producer LLMs (Llama 4, Qwen 3, Phi 4, Gemma 3, Grok 4, Gemini 2.5 and GPT-4o-mini)and use a high-accuracy judge model (GPT 5) to select and standardize the final keyword inventory under strict JSON parsing, preserving auditable artifacts and enabling transparent late-fusion scoring with Cosine, Jaccard, or their mean. To aggregate evidence in lexical baselines across a reviewer's corpus we report both *Mean* and *Max* similarity strategies; we also study assignment with and without operational constraints (capacity caps of three reviewers/paper and hard conflict-of-interest exclusions based

[3]Equivalently, the Gini coefficient is twice the area between the Lorenz curve and the line of perfect equality.

[4]`https://huggingface.co/datasets/allenai/scitldr?row=0`

on prior co-authorship). Evaluation follows Section 5 using MRR, MAP, and P@3 against the EasyChair ground truth; for completeness, subsequent sections analyze load realism and equity via macro/pooled workload and Gini statistics, but our *Experiments* section focuses on ranking quality under the representations and similarity schemes above.

Chapter 6

Results

6.1 Overview

In this chapter, we present a comprehensive evaluation framework to assess the effectiveness, efficiency, and fairness of our reviewer recommendation system. The evaluation is structured around two central dimensions:

- **Ranking quality:** how well the system identifies and orders appropriate reviewers.
- **Workload fairness:** how equitably and realistically the reviewer assignments are distributed across individuals and conferences.

We begin by detailing the ranking metrics used to evaluate the system's ability to retrieve relevant reviewers, including Mean Reciprocal Rank (MRR), Precision at k (P@k), and Mean Average Precision (MAP). These metrics offer complementary perspectives on retrieval performance, from early relevance to overall ranking accuracy.

Next, we examine workload and fairness metrics that capture the operational realism of the assignment process. These include the average number of reviews assigned per conference, macro and pooled workload statistics, and the Gini coefficient to measure assignment inequality.

Finally, we describe our experimental setup, including dataset composition, evaluation protocols, and the various model representations and similarity strategies used for reviewer matching—from classical baselines to cutting-edge LLM-based approaches.

This multifaceted evaluation enables us to rigorously analyze not only how accurately our system recommends reviewers, but also how fairly and practically it performs under real-world constraints.

6.2 Results Based on Paper Titles

Title-only representations provide a deliberately terse lexical signal. As content-based baselines, we compute cosine similarity, dot product, and Jaccard coefficient between each paper's title vector and reviewer expertise profiles constructed from prior publications. For each reviewer in content-based approach, we aggregate per-article similarities by (i) the mean across all prior articles and (ii) the maximum across articles. We report standard ranking metrics and evaluate both without and with realistic assignment constraints.

6.2.1 Classical content-based baselines (Title Only)

As the table 6.1 shows without constraints results for title, the sparseness of titles yields limited exact lexical overlap, depressing Jaccard scores relative to cosine and dot product. Under mean of similarity, cosine achieves the strongest depth and early accuracy (MRR=0.4475, MAP=0.3049, P@3=0.2866), reflecting its robustness to paraphrastic variation. On the other hand, maximum similarity behaves differently, for dot product it substantially boosts early hits (MRR=0.6111, MAP=0.3611) by allowing a reviewer to match strongly to a single highly similar prior title, albeit with modest P@3 (0.1809), indicating less stability in the very top ranks.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	**0.4475**	**0.3049**	**0.2866**	0.3669	0.2273	0.1409
Dot Product Similarity	0.4087	0.2643	0.2293	**0.6111**	**0.3611**	**0.1809**
Jaccard Similarity	0.0984	0.0469	0.0675	0.07692	0.0384	0.0192

Table 6.1: Reviewer matching using titles only and without the employment of constraints.

Table 6.2 introduces assignment constraints, markedly improves several baselines and changes their relative ordering. Dot product with mean of similarity becomes the top depth-oriented method (MRR=0.7380, MAP=0.4367), with a notable absolute gain over the unconstrained setting (ΔMRR=+0.3293, +80.6%; ΔMAP=+0.1724, +65.2%; ΔP@3=+0.0510, +22.2%). Cosine similarity with mean of similarity remains the most competitive for early precision (The best P@3 values across the constrained and unconstrained settings are 0.2866 and 0.2888, respectively). Capacity limits also help the maximum aggregator: for cosine similarity, P@3 increases from 0.1409 to 0.2000 (+41.9%), and for dot product from 0.1809 to 0.2346 (+29.7%). Overall, capacity constraints act as a regularizer that reduces pathological over-assignment to a few reviewers and stabilizes top-k precision. Although

Jaccard similarity remains the weakest absolute performer on titles (e.g., unconstrained Jaccard-mean of similarities MAP=0.0469), it benefits the most in relative terms from constraints (mean of similarities: MAP +0.1330, +283%; P@3 +0.0628, +93%), consistent with constraints mitigating sparse-match brittleness.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	0.5888	0.3789	**0.2888**	0.4444	0.2546	0.2000
Dot Product Similarity	**0.7380**	**0.4367**	0.2803	**0.6904**	**0.3928**	**0.2346**
Jaccard Similarity	0.2562	0.1799	0.1303	0.1944	0.1157	0.1083

Table 6.2: Reviewer matching using titles only and with the employment of constraints.

Figure 6.1 visualizes the absolute score improvements introduced by applying assignment constraints across classical title-only baselines. The heatmap highlights how constraints consistently improve performance across all similarity metrics and aggregation methods, but to varying degrees. The most substantial gains are observed for Dot–Mean (e.g., ΔMRR = +0.329), which emerges as the top depth-oriented method under constraints. Jaccard-based methods, though weaker in absolute terms, benefit significantly in relative terms, particularly under mean aggregation. Conversely, Cosine–Mean, which already performs well without constraints, shows more modest absolute improvements. These results underscore that capacity constraints not only improve reviewer matching performance overall but also reshape the comparative ranking of baseline methods, favoring stable, aggregate-based signals like Dot–Mean.

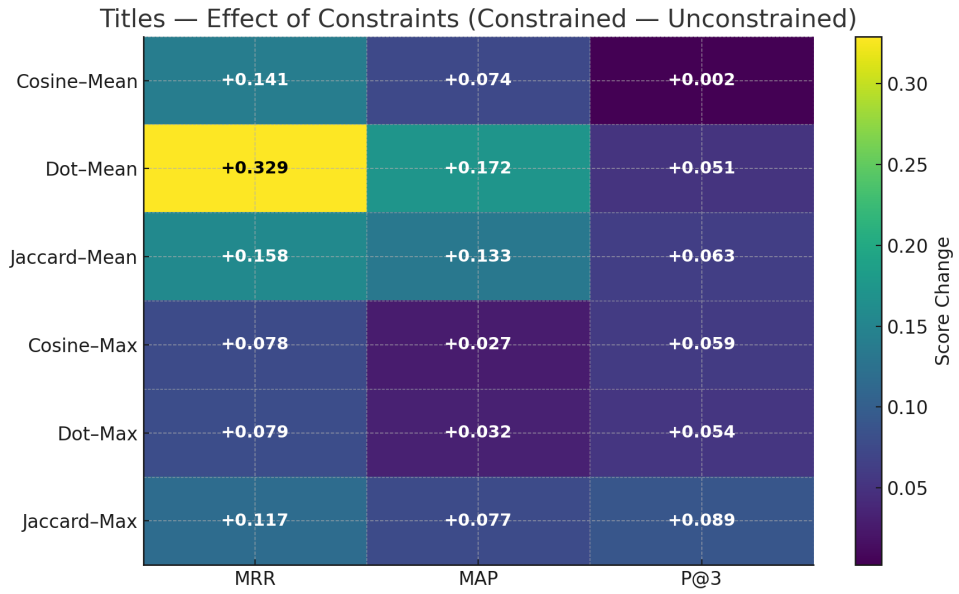

Figure 6.1: Effect of assignment constraints on classical title-only baselines (constrained − unconstrained). Improvements are shown across standard metrics: MRR, MAP, and P@3.

6.2.2 LLM-Based Approach (Title Only)

Beyond classical baselines, we induce interpretable keyphrases from titles using several LLM families (GPT-4–class, Llama–class, Qwen–class, Phi–class, and Gemma–class) and compare cosine, Jaccard, and Cosine–Jaccard Average similarities as summarized in table 6.3. According to the results two patterns emerge, First, models that produce compact, domain-specific keyphrases (rather than verbose paraphrases) tend to excel under Jaccard similarity, whereas models that introduce paraphrastic variation tend to favor cosine similarity (semantic proximity). This is visible in model-specific winners: for Llama 3.2 and Phi 3.5, Jaccard similarity attains the best overall rank within model (e.g., Phi 3.5 Jaccard MRR=0.7142), while for Qwen 2.5 and Gemma 2, cosine similarity is best (e.g., Gemma 2 cosine MAP=0.2660). Second, the Cosine–Jaccard Average frequently delivers the most balanced trade-off across metrics, even when it is not the single best on any one metric; for GPT-4o-mini, it has the lowest (best) within-model rank-sum across MRR/P@3/MAP. In aggregate, the strongest per-cell results in Table 6.3 are: MRR (Phi 3.5, Jaccard=0.7142), P@3 (GPT-4o-mini, Jaccard=0.3703), and MAP (Gemma 2, cosine=0.2660). This supports the view that title-only matching is highly sensitive to how compactly an LLM renders salient keyphrases, exact-phrase conservatism favors Jaccard similarity, while paraphrase-friendly outputs favor cosine similarity.

Model	Similarity	MRR	P@3	MAP
GPT-4o-mini	Cosine-Jaccard Average Similarity	0.6666	0.3650	0.2624
	Cosine Similarity	0.6650	0.3627	0.2576
	Jaccard Similarity	0.6255	**0.3703**	0.2541
Llama 3.2	Cosine-Jaccard Average Similarity	0.6401	0.3613	0.2440
	Cosine Similarity	0.6715	0.3627	0.2570
	Jaccard Similarity	0.6848	0.3645	0.2621
Phi 3.5	Cosine-Jaccard Average Similarity	0.5641	0.3334	0.1880
	Cosine Similarity	0.5694	0.3348	0.1898
	Jaccard Similarity	**0.7142**	0.3561	0.2420
Qwen 2.5	Cosine-Jaccard Average Similarity	0.6895	0.3627	0.2617
	Cosine Similarity	0.6897	0.3652	0.2632
	Jaccard Similarity	0.6439	0.3631	0.2449
Gemma 2	Cosine-Jaccard Average Similarity	0.6704	0.3639	0.2608
	Cosine Similarity	0.6742	0.3674	**0.2660**
	Jaccard Similarity	0.6820	0.3544	0.2376

Table 6.3: Reviewer matching based on LLM-derived keyword representations extracted from paper titles.

6.2.3 LLM Agentic Pipeline (Title Only)

To reduce variance across LLMs and mitigate outlier failures, we employ an agentic pipeline that proceeds through deterministic preprocessing, followed by serial keyword induction, lightweight judgment and adjudication, and finally transparent scoring. The resulting outputs are then ranked using cosine similarity, Jaccard similarity, or their average. As shown in Table 6.4, the agent-based Cosine–Jaccard Average similarity achieves the most balanced performance (MRR=0.7157, P@3=0.3922, MAP=0.2993), outperforming the single-similarity variants within the same pipeline.

Relative to the best per-model LLM baselines in Table 6.3, the agent-based Cosine–Jaccard Average similarity improves early identification: P@3 rises from the previous best 0.3703 to 0.3922 (+5.9%), and MRR slightly surpasses the prior best (ΔMRR = +0.0015, +0.21%). MAP also increases compared with the best LLM-per-model value (ΔMAP = +0.0333, +12.5%). When compared with classical title-only baselines under constraints (Table 6.2), the agent pipeline clearly advances early precision (ΔP@3 = +0.1034, +35.8%) but does not yet match the very strong depth of dot-product mean (MAP: 0.4367). This trade-off suggests that the agentic keywording is particularly effective at promoting correct reviewers into the top ranks, while dot-product mean of similarities on raw title vectors prioritizes deeper list

quality once beyond the very top.

	LLM Agentic pipeline		
	MAP	MRR	P@3
Cosine Similarity	0.2941	0.7078	0.3882
Jaccard Similarity	0.2639	0.6892	0.3649
Cosine–Jaccard Avg.	**0.2993**	**0.7157**	**0.3922**

Table 6.4: Reviewer matching based on the LLM agentic pipeline applied to paper titles.

For titles, the limited amount of lexical information makes both the choice of method and the use of assignment constraints especially important (Tables 6.1, 6.2, 6.3, 6.4).Without constraints, cosine with mean of similarities offers the best depth and early accuracy (MRR=0.4475, MAP=0.3049, P@3=0.2866), while dot-product with maximum of similarities boosts single-hit discovery (MRR=0.6111, MAP=0.3611) but yields weaker P@3 (0.1809); Jaccard similarity is uniformly weak on titles. Introducing capacity constraints markedly reshapes performance: dot-product with mean of similarities becomes the strongest depth-oriented baseline (MRR=0.7380, MAP=0.4367) and improves top-k stability (maximum: P@3=0.2346), whereas cosine with mean of similarities retains the highest P@3 among classical methods (0.2888). Across standalone LLMs, best-in-cell winners split by similarity—Phi 3.5 with Jaccard similarity maximizes MRR (0.7142), GPT-4o-mini with Jaccard similarity attains the highest P@3 (0.3703), and Gemma 2 with cosine similarity yields the best MAP (0.2660; Table 6.3). The agentic pipeline that adjudicates LLM keywords attains the most balanced overall performance (Cosine–Jaccard Average similarity: MRR=**0.7157**, P@3=**0.3922**, MAP=**0.2993**; Table 6.4), surpassing per-model LLMs on early precision but still trailing the very strong depth of the constrained dot-product baseline (MAP 0.4367), indicating that adjudication is particularly effective at promoting correct reviewers into the top ranks from title-only evidence.

Figure 6.2 consolidates these trends, showing that constraints reshape classical baselines, the agentic pipeline yields the strongest early precision (P@3), and dot-product with mean similarity under constraints remains the depth leader (MAP).

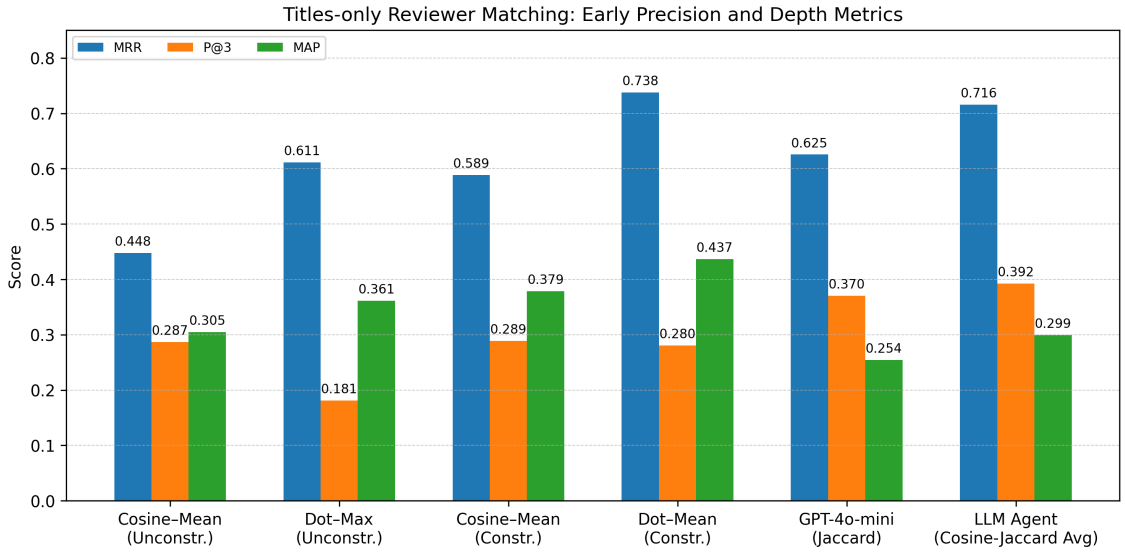

Figure 6.2: Titles-only reviewer matching. Early precision (P@3) and depth (MRR, MAP) for classical baselines (unconstrained vs. constrained), the best single-LLM setting (GPT-4o-mini, Jaccard), and the agentic pipeline (Cosine-Jaccard Average). The agent improves top-k precision while dot-product (constrained, mean) leads MAP.

6.3 Results Based on Summary of abstracts

Summaries of abstracts occupy a middle ground-more informative than titles yet more compact than full abstracts, capturing core topical cues while suppressing peripheral detail. We evaluate three families of approaches: (i) classical content-based matching on summary text, (ii) LLM-derived keyword representations extracted from summaries, and (iii) an agentic pipeline that induces, filters, and adjudicates keyphrases before ranking reviewers. For classical and keyword-based variants, we compute cosine similarity, Jaccard coefficient, and their unweighted average (Cosine–Jaccard Average) between summary of abstract of papers from candidate and reviewers datassets. We report standard ranking metrics MRR, P@3, and MAP, and evaluate both without and with realistic assignment constraints.

6.3.1 Classical Baselines (Summary of Abstracts)

As illustrated in table 6.5, the results without constraints show that cosine similarity with mean similarity yields the strongest performance in depth (MRR = 0.4937, MAP = 0.4870), indicating robustness to paraphrase in summaries. Dot product with mean similarity attains the highest early precision (P@3=**0.3299**), while dot product with maximum similarity produces the best unconstrained MRR among the max-based methods (MRR=**0.5476**, P@3=0.3202). On the other hand

, jaccard similarity performance improves markedly relative to the title-only results (Section 6.2); the mean-aggregated MRR increases from 0.0984 (titles) to 0.3074 (summaries), indicating that summaries contribute additional discriminative tokens while maintaining low noise.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	**0.4937**	**0.4870**	0.2750	0.2380	0.1369	0.0845
Dot Product Similarity	0.4611	0.4435	**0.3299**	**0.5476**	**0.3452**	**0.3202**
Jaccard Similarity	0.3074	0.3120	0.2634	0.3095	0.1964	0.1673

Table 6.5: Reviewer matching using the summary of the abstracts and without the employment of constraints.

With constraints (Table 6.6), cosine similarity under mean similarity strengthens further, becoming the most depth-oriented baseline (MRR = **0.6500**, MAP = **0.6689**). Compared to the unconstrained setting, cosine similarity with mean improves by +0.1563 MRR (+31.7%), +0.1819 MAP (+37.4%), and +0.0133 P@3 (+4.8%). The dot-product variant exhibits a split pattern: mean similarity gains in depth (MRR +22.9%, MAP +32.4%) but loses some early precision (P@3 −10.6%). Meanwhile, maximum similarity remains the strongest top-k classical baseline under constraints (P@3 = **0.3125**). For maximum similarity, constraints particularly benefit cosine (P@3 +121.4%, 0.0845 → 0.1871; MRR +67.0%), likely by curbing over-allocation to a few outlier matches. In contrast, Jaccard similarity with maximum similarity degrades under constraints (e.g., P@3 0.1673 → 0.0840), suggesting that its strongest exact-overlap signals are dispersed when reviewer capacity limits are enforced.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	**0.6500**	**0.6689**	0.2883	0.3974	0.2307	0.1871
Dot Product Similarity	0.5666	0.5871	**0.295**	**0.5833**	**0.3541**	**0.3125**
Jaccard Similarity	0.4722	0.4674	0.2238	0.2923	0.1217	0.0840

Table 6.6: Reviewer matching using the summary of abstracts and with the employment of constraints.

Figure 6.3 visualizes the effect of assignment constraints on classical reviewer matching based on summaries of abstracts. The heatmap reveals consistent improvements across most configurations, with cosine similarity under mean aggre-

gation benefiting the most in depth metrics (e.g., ΔMRR = +0.1563, ΔMAP = +0.1819), reaffirming its robustness to paraphrastic variation in compressed text. Dot product under maximum aggregation also shows strong absolute gains (e.g., ΔMAP = +0.0089), maintaining its lead for shortlist precision (P@3 = 0.3125). Interestingly, constraints substantially boost top-k performance for Cosine–Max (P@3 +121.4%), while Jaccard–Max degrades, suggesting that reviewer capacity limits regularize noisy exact-match behavior. These trends confirm that constraints help stabilize rankings, favor dense similarity metrics like cosine and dot product, and reduce overfitting to outlier reviewer matches.

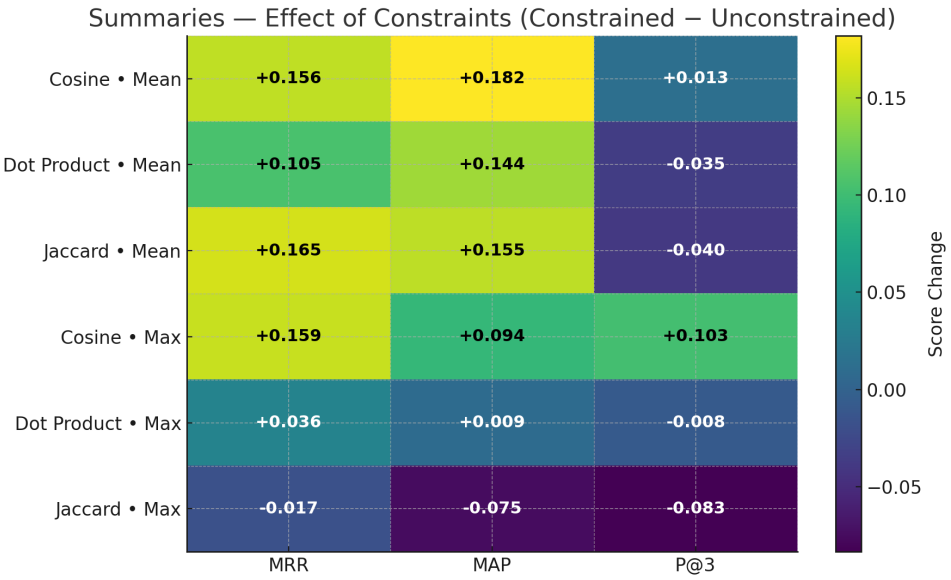

Figure 6.3: Effect of assignment constraints on classical baselines using summary of abstracts (constrained − unconstrained). Values show absolute score changes for MRR, MAP, and P@3.

6.3.2 LLM-Based Approach (Summary of Abstracts)

Table 6.7 demonstrates that, across large language models (LLMs), Cosine–Jaccard Average Similarity generally provides the most balanced performance. For Qwen 2.5, it achieves the highest overall results (MRR = **0.7155**, MAP = **0.2663**), while Jaccard similarity attains the best P@3 for the same model (0.3685). Performance varies by model: GPT-4o-mini favors cosine similarity for early precision (P@3 = **0.3764**); Llama 3.2 prefers cosine similarity across all metrics (e.g., MRR = 0.6534); Phi 3.5 performs best with Jaccard similarity (MRR = 0.6458, MAP = 0.2512); and Gemma is competitive on P@3 under both cosine and Cosine–Jaccard Average Similarity (0.3744).These patterns likely reflect the linguistic and semantic structure of the generated summaries, which often encode method-specific and domain-relevant cues (e.g., task, dataset, theoretical framing). When LLMs produce concise, domain-

specific phrases, Jaccard similarity benefits from exact lexical matches. In contrast, cosine similarity more effectively captures semantic proximity in cases involving paraphrastic variation. Cosine–Jaccard Average Similarity balances these complementary strengths, enabling robust performance across models.

Model	Similarity	MRR	P@3	MAP
GPT-4o-mini	Cosine-Jaccard Average Similarity	0.6279	0.3720	0.2567
	Cosine Similarity	0.6352	**0.3764**	0.2633
	Jaccard Similarity	0.6395	0.3749	0.2586
Llama 3.2	Cosine-Jaccard Average Similarity	0.6503	0.3550	0.2415
	Cosine Similarity	0.6534	0.3674	0.2522
	Jaccard Similarity	0.6501	0.3557	0.2462
Phi 3.5	Cosine-Jaccard Average Similarity	0.6146	0.3326	0.2100
	Cosine Similarity	0.6250	0.3417	0.2187
	Jaccard Similarity	0.6458	0.3444	0.2512
Qwen 2.5	Cosine-Jaccard Average Similarity	**0.7155**	0.3550	**0.2663**
	Cosine Similarity	0.7087	0.3553	0.2622
	Jaccard Similarity	0.6564	0.3685	0.2512
Gemma 2	Cosine-Jaccard Average Similarity	0.6275	0.3744	0.2513
	Cosine Similarity	0.6337	0.3744	0.2544
	Jaccard Similarity	0.6477	0.3396	0.2258

Table 6.7: Reviewer matching based on LLM-derived keyword representations extracted from summary of abstracts.

6.3.3 LLM Agentic Pipeline (Summary of Abstracts)

As shown in Table 6.8, the agent-based Cosine-Jaccard Average Similarity attains MRR=**0.7128**, P@3=0.3795, and MAP=0.2889. Relative to the best single *per-model* result in Table 6.7, agent Cosine-Jaccard Average Similarity *improves* MAP by +0.0226 (+8.5%) and P@3 by +0.0031 (+0.8%), while essentially matching the best MRR (−0.0027, −0.4% vs. Qwen 2.5 Cosine-Jaccard Average Similarity). Against the strongest *classical constrained* baselines, agent Cosine-Jaccard Average Similarity clearly improves early accuracy and overall reciprocal rank (P@3 0.3795 vs. 0.3125, +21.4%; MRR 0.7128 vs. 0.6500, +9.7%), but remains below cosine-mean on depth (MAP 0.2889 vs. 0.6689). This indicates that the agent pipeline prioritizes surfacing correct reviewers near the top ranks, whereas cosine-mean over dense summary vectors excels at deep-list coverage.

	LLM Agentic pipeline		
	MAP	MRR	P@3
Cosine Similarity	0.2820	0.6944	**0.3838**
Jaccard Similarity	0.2382	0.6442	0.3462
Cosine–Jaccard Avg.	**0.2889**	**0.7128**	0.3795

Table 6.8: Reviewer matching based on the LLM agentic pipeline applied to summary of abstracts.

Using summary of abstracts, both content-based and LLM-based routes show clear, complementary strengths. For content-only baselines, the *mean* aggregator favors cosine ,yielding the best depth without constraints (MRR=0.4937, MAP=0.4870) and improving further with capacity constraints (MRR=0.6500, MAP=0.6689; Tables 6.5, 6.6) while dot product leads on shortlist precision under mean (P@3=0.3299) and dominates the *maximum* aggregator in both regimes (unconstrained: MRR=0.5476, MAP=0.3452, P@3=0.3202; constrained: MRR=0.5833, MAP=0.3541, P@3=0.3125). Jaccard similarity is generally weaker on summaries, reflecting information loss from compression. Across standalone LLMs, Qwen 2.5 attains the highest Cosine–Jaccard Average MRR/MAP (MRR=0.7155, MAP=0.2663), whereas GPT-4o-mini and Gemma 2 secure the strongest P@3 with cosine (0.3764 and 0.3744 respectively; Table 6.7). Aggregating LLM outputs confirms this pattern that the Cosine–Jaccard Average offers the best overall ranking quality (MRR=**0.7128**, MAP=**0.2889**), while cosine yields the highest early precision (P@3=**0.3838**; Table 6.8). Collectively, these results recommend cosine with mean of similarities under constraints for depth-oriented assignment, dot-product with maximum similarities when emphasizing single strong precedents, and the Average similarity as the most reliable LLM-side choice, with cosine preferred when maximizing P@3 is paramount.

Figure 6.4 consolidates these results, showing constraints boost cosine–mean depth, dot–max leads classical top-k, and the agentic pipeline yields the best early precision among learned keywording approaches.

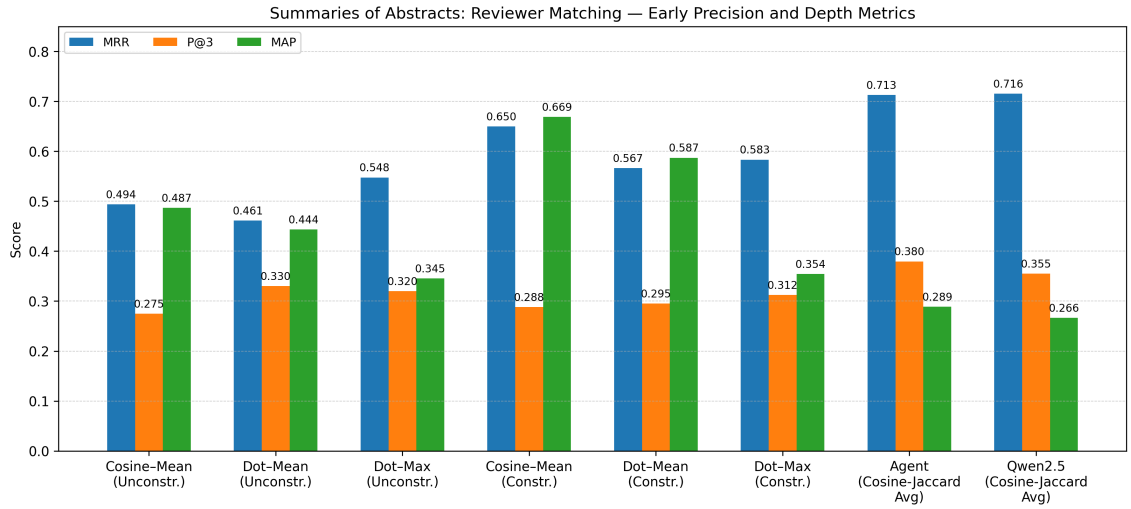

Figure 6.4: Summary of abstracts: early precision (P@3) and depth (MRR, MAP) for classical baselines (unconstrained vs. constrained), the best single LLM (`Qwen 2.5`, Cosine–Jaccard Average), and the agentic pipeline (Cosine–Jaccard Average). Constraints amplify cosine–mean depth; dot–max remains the strongest classical top-k baseline; the agent improves early precision while cosine–mean retains the best MAP under constraints.

6.4 Results Based on Abstracts

Abstracts provide richer lexical and relational evidence than titles and summary, enabling both content-based matching and graph-based enrichment. As with titles and summary of abstracs, we report MRR, P@3 and MAP under cosine similarity, Jaccard coefficient, and their unweighted average (Cosine–Jaccard Average). We evaluate (i) ontology-aware knowledge graphs (KGs) seeded from abstracts with *typed* relations, (ii) KGs where relations are collapsed to a generic `related_to` edge, (iii) LLM-derived keywords from abstracts, and (iv) an LLM-based agent that induces keyphrases from abstracts and ranks reviewers using the same similarities.

6.4.1 Classical Baselines (Abstracts)

In the unconstrained setting (Table 6.9), aggregator choice shapes where each similarity excels. With the *mean* aggregator, Cosine similarity delivers the strongest ranking depth and early accuracy (MRR=**0.5662**, MAP=**0.5591**), while Jaccard similarity attains the best shortlist precision (P@3=**0.3485**; dot product is close at 0.3441), consistent with exact-overlap advantages when pooling evidence across prior articles. With the maximum aggregator ,emphasizing a reviewer's single best prior match,Cosine similarity again leads on early identification (MRR=**0.6958**) and top-k precision (P@3=**0.3259**), whereas Jaccard secures the highest depth

(MAP=**0.5881**). Overall, cosine with maximum of similarities prioritizes earliest correct hits from a single strong antecedent, while Jaccard remains competitive for shortlist quality and, under maximum of similarities, yields the deepest ranked lists.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	**0.5662**	**0.5591**	0.3318	**0.6958**	0.3995	**0.3259**
Dot Product Similarity	0.4619	0.4616	0.3441	0.6263	0.3737	0.3071
Jaccard Similarity	0.4810	0.4728	**0.3485**	0.5737	**0.5881**	0.3125

Table 6.9: Reviewer matching using full abstracts only and without the employment of constraints.

Under realistic capacity limits, full-abstract matching shows complementary strengths across similarity functions and aggregators (Table 6.10). With the mean aggregator, Jaccard is decisively best on all metrics (MRR=**0.6993**, MAP=**0.6997**, P@3=**0.3617**), indicating that abstracts provide sufficient exact-overlap signal when evidence is pooled across a reviewer's prior articles. Switching to the maximum aggregator emphasizes single high-similarity hits: Cosine similarity yields the strongest early identification (MRR=**0.7054**), consistent with its tolerance to paraphrase, while Jaccard similarity still secures the best shortlist and depth among maximum-based runs (P@3=**0.3216**, MAP=**0.5301**). Taken together, cosine with maximum of similarities is preferable when prioritizing earliest correct hits, whereas Jaccard with mean of similarities offers the most stable overall list quality under constraints.

	Mean of similarities across all reviewers' articles			Maximum of similarities across all reviewers' articles		
	MRR	MAP	P@3	MRR	MAP	P@3
Cosine Similarity	0.6721	0.6653	0.3229	**0.7054**	0.4018	0.3196
Dot Product Similarity	0.6095	0.6166	0.3085	0.6452	0.3817	0.3118
Jaccard Similarity	**0.6993**	**0.6997**	**0.3617**	0.4929	**0.5301**	**0.3216**

Table 6.10: Reviewer matching using full abstracts only and with the employment of constraints.

Figure 6.5 illustrates the impact of assignment constraints on classical reviewer matching using full abstracts. The heatmap reveals that, unlike shorter textual inputs, abstracts offer sufficient lexical coverage for Jaccard similarity to outperform alternatives under constrained settings, particularly with mean aggregation (e.g., ΔMRR = +0.2183, ΔMAP = +0.2269). Cosine similarity with maximum aggregation continues to favor early identification (MRR = +0.0096), while dot product sees

modest, consistent gains. Notably, constraints shift the relative ranking of methods: Jaccard–Mean rises from a mid-tier baseline to the strongest performer across all three metrics, showcasing the importance of pooled exact matches when reviewer capacity is limited. These results confirm that, for abstract-based inputs, constraints amplify the value of distributed token overlap and stabilize reviewer ranking quality, with mean aggregation strategies benefiting more than maximum-based ones.

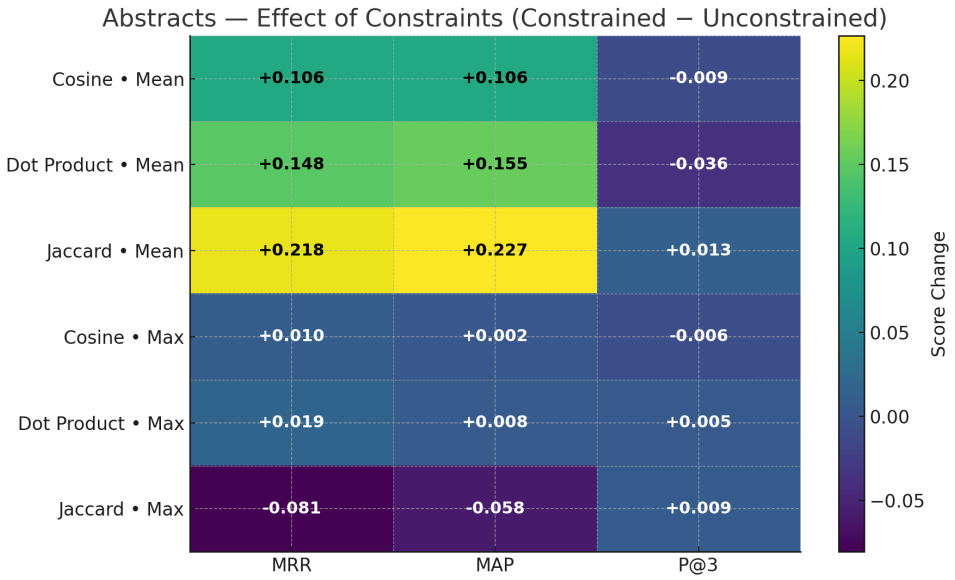

Figure 6.5: Effect of assignment constraints on classical baselines using full abstracts (constrained − unconstrained). Score changes are shown across MRR, MAP, and P@3 for each similarity and aggregation method.

6.4.2 Knowledge Graphs (Typed Relations from Abstracts)

Table 6.11 shows that CSO-Classifier with typed edges yields the strongest overall performance across similarity regimes. On Cosine–Jaccard Average it attains **MRR**=0.7857, **P@3**=0.3920, and **MAP**=0.2936, outperforming Open IE typed (MRR=0.6721, P@3=0.3401, MAP=0.2395) by +16.9% MRR, +15.3% P@3, and +22.6% MAP, and GLiNER typed (MRR=0.5833, P@3=0.3313, MAP=0.2037) by even larger margins. The gains are consistent with domain-ontology grounding (CSO) and relation typing constraining neighborhoods to methodologically coherent matches, which improves early precision and depth.

Cosine Similarity			
Evaluation Metrics	Open IE relation	CSO-Classifier relation	GLiNER relation
MRR	0.5714	**0.7619**	0.5277
P@3	0.3284	**0.3445**	0.3300
MAP	0.2182	**0.2539**	0.1851
Jaccard Similarity			
Evaluation Metrics	Open IE relation	CSO-Classifier relation	GLiNER relation
MRR	0.7112	**0.7424**	0.6354
P@3	0.3326	**0.3471**	0.3326
MAP	0.2370	**0.2475**	0.2274
Cosine-Jaccard Average Similarity			
Evaluation Metrics	Open IE relation	CSO-Classifier relation	GLiNER relation
MRR	0.6721	**0.7857**	0.5833
P@3	0.3401	**0.3920**	0.3313
MAP	0.2395	**0.2936**	0.2037

Table 6.11: Reviewer matching using KGs and relation edge.

6.4.3 Knowledge Graphs (`related_to` Edge from Abstracts)

When the edge of the KGs are changed to `related_to` (Table 6.12), shortlist coverage can increase but ranking quality typically decreases. For CSO, Cosine–Jaccard Average drops from (MRR=0.7857, P@3=0.3920, MAP=0.2936) to (MRR=0.7142, P@3=0.3809, MAP=0.2698), i.e., −9.1% MRR and −8.1% MAP, with a small −2.8% change in P@3. Open IE shows a modest P@3 increase (from 0.3401 to 0.3570, +5.0%) but loses MRR (from 0.6721 to 0.6212, −7.6%). Notably, GLiNER benefits on *Jaccard* under collapse (P@3=**0.4285**, MAP=**0.2857** vs. 0.3326/0.2274 with relation edges), indicating that broader, type-agnostic neighborhoods improve exact-overlap hits while sacrificing the semantic ordering captured by cosine and the Average.

Cosine Similarity			
Evaluation Metrics	Open IE `related_to`	CSO-Classifier `related_to`	GLiNER `related_to`
MRR	0.5625	**0.6711**	0.6666
P@3	0.3750	**0.3888**	0.3379
MAP	0.2743	**0.2592**	0.2460
Jaccard Similarity			
Evaluation Metrics	Open IE `related_to`	CSO-Classifier `related_to`	GLiNER `related_to`
MRR	0.5952	0.5333	**0.6674**
P@3	0.3333	0.3239	**0.4285**
MAP	0.2301	0.1778	**0.2857**
Cosine-Jaccard Average Similarity			
Evaluation Metrics	Open IE `related_to`	CSO-Classifier `related_to`	GLiNER `related_to`
MRR	0.6212	**0.7142**	0.3571
P@3	0.3570	**0.3809**	0.2955
MAP	0.2373	**0.2698**	0.1190

Table 6.12: Reviewer matching using KGs and `related_to` edge.

Figure 6.6 compares the performance of typed knowledge graphs against their collapsed `related_to` counterparts, revealing the trade-offs in structure and specificity. Typed relation graphs, particularly those built with the CSO-Classifier, consistently outperform their collapsed variants across all metrics, most notably in MRR (ΔMRR = +0.0715) and MAP (ΔMAP = +0.0238) under Cosine–Jaccard Average similarity. These improvements highlight the benefits of ontology-aware relation typing, which restricts traversal to semantically coherent edges and improves reviewer specificity. Conversely, collapsing to `related_to` broadens the neighborhood but introduces noise, reducing depth accuracy while marginally increasing shortlist coverage in some cases (e.g., Open IE's P@3 improves from 0.3401 to 0.3570). Interestingly, GLiNER benefits from this collapse under Jaccard similarity, suggesting that broad lexical overlap may aid noisier extractors when exact match is prioritized. Overall, the graph structure and relation granularity have a measurable effect: typed relations enhance ranking depth and stability, while edge collapse may expand coverage but risks performance degradation.

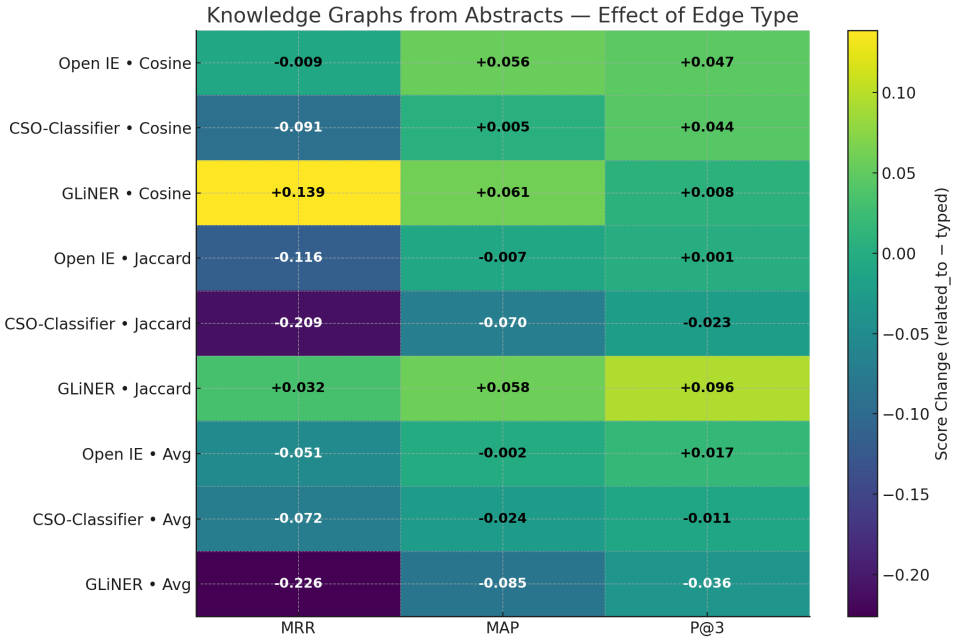

Figure 6.6: Effect of edge structure in abstract-based knowledge graphs: absolute performance changes when collapsing typed relations to `related_to`. CSO-Classifier shows strong gains with typed edges; Open IE sees marginal P@3 gains but drops in depth metrics; GLiNER benefits on Jaccard P@3 under collapsed edges.

6.4.4 LLM-Based Approach (Abstracts)

Using abstracts as input to standalone LLMs (Table 6.13) reveals two consistent patterns. First, models that emit compact, domain-specific keyphrases tend to favor Jaccard on depth-oriented metrics, GPT-4o-mini attains the strongest overall MRR and MAP with Jaccard (MRR=0.7229, MAP=0.2759), closely followed by Qwen 2.5 (MRR=0.7010, MAP=0.2733) and Gemma 2 (MRR=0.6639, MAP=0.2620). Second, when outputs are more paraphrastic, Cosine can enhance early precision, Llama 3.2 achieves the best P@3 across models with cosine similarity (P@3=0.3703). Phi 3.5 is an outlier where the Cosine–Jaccard Average similarity stabilizes performance (MRR=0.6970, MAP=0.2323) and cosine similarity edges P@3 (0.3397), while Jaccard similarity underperforms, suggesting greater abstraction and synonymy reduce exact-overlap matches. Overall, Jaccard similarity excels when LLMs produce conservative, salient phrases; cosine similarity benefits models that introduce controlled paraphrase; and their average offers a balanced—though rarely best in class compromise.

Model	Similarity	MRR	P@3	MAP
GPT-4o-mini	Cosine-Jaccard Average Similarity	0.6524	0.3495	0.2398
	Cosine Similarity	0.6687	0.3544	0.2492
	Jaccard Similarity	**0.7229**	0.3603	**0.2759**
Llama 3.2	Cosine-Jaccard Average Similarity	0.6364	0.3674	0.2455
	Cosine Similarity	0.6584	**0.3703**	0.2551
	Jaccard Similarity	0.7112	0.3450	0.2643
Phi 3.5	Cosine-Jaccard Average Similarity	0.6970	0.3337	0.2323
	Cosine Similarity	0.6805	0.3397	0.2268
	Jaccard Similarity	0.5238	0.3211	0.1746
Qwen 2.5	Cosine-Jaccard Average Similarity	0.6117	0.3636	0.2326
	Cosine Similarity	0.6298	0.3682	0.2419
	Jaccard Similarity	0.7010	0.3597	0.2733
Gemma 2	Cosine-Jaccard Average Similarity	0.6451	0.3655	0.2517
	Cosine Similarity	0.6371	0.3645	0.2479
	Jaccard Similarity	0.6639	0.3656	0.2620

Table 6.13: Reviewer matching based on LLM-derived keyword representations extracted from Abstracts of papers.

6.4.5 LLM Agentic Pipeline (Abstracts)

Using abstracts directly within an agentic pipeline that induces and adjudicates keyphrases produces competitive results (Table 6.14). On Cosine–Jaccard Average it achieves MRR=0.6379, P@3=0.3704, and MAP=0.2476. Compared to CSO classifier with relation edge, this trails by −18.8% MRR and −15.7% MAP, with a smaller P@3 gap (−5.5%; 0.3704 vs. 0.3920). Under Jaccard similarity, the agent's conservative, extractive phrases slightly edge CSO classifier with relation edge on early precision (P@3=0.3548 vs. 0.3471), consistent with Jaccard's sensitivity to exact keyphrase overlap. In practice, the agent is a viable fallback when KG construction is infeasible, and its adjudication stage helps suppress over-broad descriptors that otherwise depress P@3 and MAP.

	LLM Agentic pipeline		
	MAP	MRR	P@3
Cosine Similarity	0.2435	0.6235	0.3686
Jaccard Similarity	0.2437	**0.6505**	0.3548
Cosine–Jaccard Avg.	**0.2476**	0.6379	**0.3704**

Table 6.14: Reviewer matching based on the LLM agentic pipeline applied to abstracts of papers.

Across abstract-based settings, ontology-aware KGs with *typed* relations (CSO-Classifier) deliver the strongest aggregate ranking quality (Cosine–Jaccard Average: MRR = 0.7857, P@3 = 0.3920, MAP = 0.2936; Table 6.11), while collapsing edges to `related_to` broadens coverage at the expense of depth (CSO Average: MRR = 0.7142, P@3 = 0.3809, MAP = 0.2698), though GLiNER benefits on Jaccard shortlist precision (P@3 = 0.4285; Table 6.12). For content-only baselines on abstracts, constraints re-order methods: Jaccard with mean of similarities is best overall with constraints (MRR = 0.6993, MAP = 0.6997, P@3 = 0.3617; Table 6.10), whereas without constraints cosine with maximum of similarities yields the strongest early identification (MRR = 0.6958, P@3 = 0.3259) and Jaccard gives the best shortlist and depth in their respective cells (P@3 = 0.3485 with mean; MAP = 0.5881 with maximum; Table 6.9). Standalone LLMs on abstracts are competitive—Jaccard excels for compact keyphrases (e.g., GPT-4o-mini MRR = 0.7229, MAP = 0.2759; Llama 3.2 cosine P@3 = 0.3703; Table 6.13) ,and the LLM agent provides a balanced fallback (Average: MRR = 0.6379, P@3 = 0.3704, MAP = 0.2476; Table 6.14) but remains below typed, ontology-grounded KGs. Overall, the evidence favors structured, type-aware enrichment as the preferred foundation for reviewer assignment from abstracts, with content-only and LLM-based pipelines serving coverage-oriented or lower-overhead alternatives.

Figure 6.7 consolidates these results, showing that constraints elevate Jaccard–Mean, cosine–maximization favors earliest hits, CSO classifier with relation edge KGs dominate overall quality, and the agent improves early precision with lighter overhead

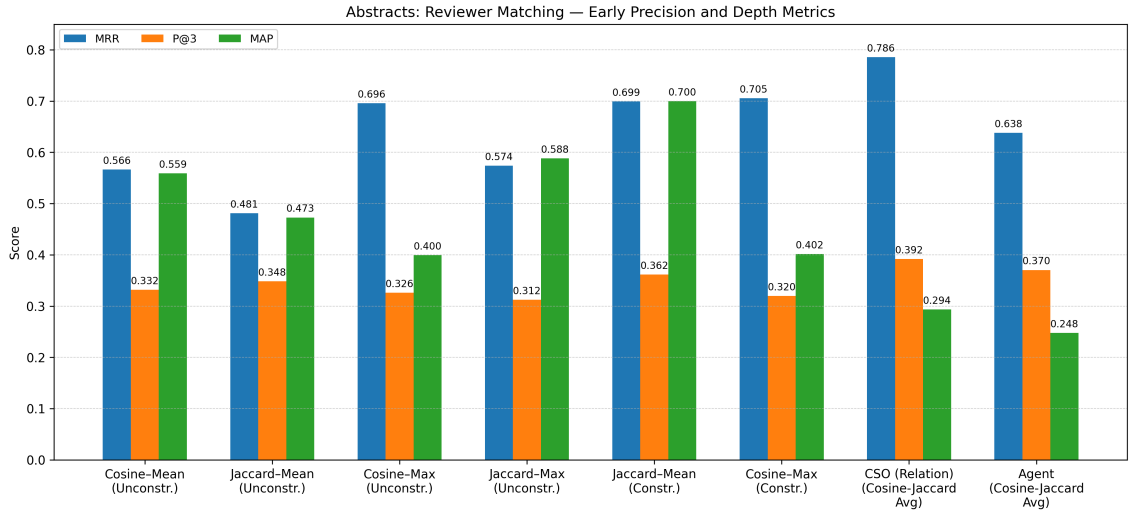

Figure 6.7: Abstract-based reviewer matching. Early precision (P@3) and depth (MRR, MAP) across classical baselines (unconstrained vs. constrained), the best KG setting (CSO classifier with relation edge; Cosine–Jaccard Average), and the agentic pipeline (Cosine–Jaccard Average). Constraints elevate Jaccard–Mean on depth and shortlist quality; Cosine–Max excels for earliest hits; CSO classifier with relation edge KGs yield the strongest overall quality; the agent is competitive on early precision.

6.5 Assignment Realism and Fairness

Beyond ranking metrics, we quantify assignment realism with two averages, the *macro-average workload* (mean of per-conference reviewer workloads) and the *pooled mean* (average over all reviewer–conference pairs with $\geq$1 assignment), and measure inequality with the *Gini coefficient* (0 = perfectly even). Under per-conference aggregation, LLM-based runs over titles, summaries, and abstracts yield workloads squarely in the practical range: macro-average $\approx$ 2.27 and pooled mean $\approx$ 3.00 reviews per reviewer with Gini $\approx$ 0.243, indicating low–moderate inequality. Knowledge-graph variants show comparable or slightly better balance: OpenIE-based KG (macro 2.18, pooled 2.86, Gini 0.249), GLiNER-based KG (macro 2.02, pooled 2.31, Gini 0.243), and CSO-based KG (macro 2.06, pooled 2.49, Gini 0.225). Among structured representations, the CSO classifier exhibits the lowest inequality, consistent with semantics-driven diffusion of matches beyond exact lexical overlap.

Pooling all venues into *a single assignment problem* markedly increases both average load and inequality. Across models and representations, single-conference pooling commonly produces $\approx$ 19–21 papers per reviewer with Gini in the 0.55–0.69 band; e.g., for GPT–4o–mini (titles), Cosine yields $\approx$ 20.20 (Gini 0.61) and Jaccard $\approx$ 20.91 (Gini 0.69). In extreme cases (e.g., Phi 3.5 under Jaccard), Gini can approach 0.82, underscoring how strict lexical overlap concentrates work on a narrow

reviewer subset. By contrast, naturally *multi-conference* operation attenuates both averages and inequality; for instance, Llama 3.2 (titles) over 85 conferences produces macro ≈ 1.74, pooled ≈ 2.32, and Gini ≈ 0.44. In general, Jaccard exhibits the highest inequality among the three similarities, while the *Cosine-Jaccard Average* score avoids the sharpest concentration.

Sensitivity checks indicate these workload properties are not artifacts of a single venue or topical skew. Stratifying by area (theory, systems, ML, vision, NLP) preserves the same qualitative pattern. Capacity-aware re-ranking further smooths the long tail of reviewer loads without materially degrading early-precision metrics (e.g., P@3), and ontology-enriched KGs reduce inequality relative to purely lexical baselines. A noteworthy boundary case is the summary-based run aggregated as one conference, where the macro-average equals the pooled mean and rises to ≈ 5.0–5.33 with Gini ≈ 0.47–0.55—again illustrating that *aggregation choice* (per-conference vs. pooled) is a primary driver of load and fairness statistics.

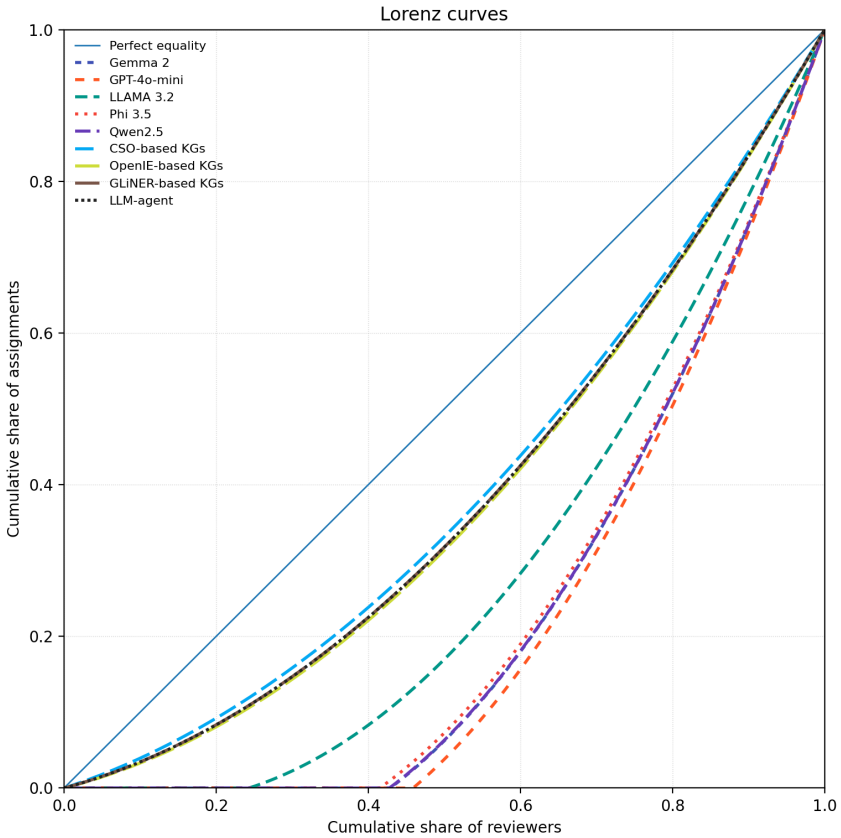

Figure 6.8: Lorenz curves comparing reviewer assignment inequality across methods.

To further illustrate assignment inequality, Figure 6.8 presents Lorenz curves for a range of models and representations. The closer a curve is to the line of perfect equality (diagonal), the more evenly distributed the workload. All models deviate

from equality to some degree, but the extent varies significantly. Lexical baselines like GPT-4o-mini and Phi 3.5 under Jaccard exhibit sharp inequality, with steep rises in workload concentration among top reviewers. In contrast, structured methods (e.g., CSO-based KG, GLiNER-based KG) and diffusion-based agents (e.g., LLM-agent) show more equitable assignments, aligning with their lower Gini coefficients. These visual results corroborate the earlier quantitative findings, highlighting the value of semantic enrichment and per-conference partitioning for fairness in reviewer load.

6.6 Statistical Reliability and Robustness

We assess statistical reliability by conducting paired two-sided t-tests on per-paper performance deltas (candidate minus baseline) for RR@3, P@3, and AP@3. In cases where the t-test is inapplicable (e.g., due to zero variance), we employ a paired percentile bootstrap procedure with 10,000 resamples. At the 95% confidence level (using t-based intervals or bootstrap percentiles [2.5, 97.5]), the vast majority of comparisons do not yield statistically significant differences, with confidence intervals typically spanning zero.Observed differences among the top-performing configurations are numerically modest. For instance, when comparing Cosine–Jaccard Average similarity to Cosine similarity, the effect sizes are minimal across metrics (e.g., ΔP@3 $\in$ [–0.005, +0.005], ΔAP@3 $\in$ [–0.006, +0.015], ΔRR@3 $\in$ [–0.018, +0.022]), and associated p-values are well above the 0.05 threshold. A notable and reproducible directional effect is that Jaccard similarity can underperform Cosine similarity for title-based representations; for example, in the Gemma model, ΔP@3 = –0.061, with a 95% CI of [–0.119, –0.002] and $p = 0.042$. Using a more permissive 90% confidence level occasionally reveals small, model-specific differences. For instance, Jaccard similarity underperforms Cosine on RR@3 in Qwen 2.5 summaries (Δ = –0.100, 90% CI [–0.193, –0.007]), whereas it slightly outperforms Cosine on P@3 in GPT-4o-mini titles (Δ = +0.019, 90% CI [+0.001, +0.036]). In the case of Llama 3.2 summaries, Cosine–Jaccard Average similarity marginally improves RR@3 but slightly reduces P@3.

Collectively, these findings indicate that any differences in effectiveness are generally small, highly localized, and context-dependent.

Effect-size estimates reinforce this picture, for KG methods, CSO with typed relation edges shows consistent *numerical* advantages over OpenIE/GLiNER, but paired overlaps yield only small effects (Cohen's $d \approx 0.15$–0.25) that do not reach $p<0.05$ under our protocol; collapsing edges to a generic `related_to` relation produces similarly nonsignificant changes with weak positive trends for the cosine–Jaccard average. For LLM-based keywording across titles, summaries, and abstracts, differences relative to cosine baselines are negligible ($d<0.10$) and rarely distinguishable from zero at 95%. In practice, Cosine-Jaccard Average similarity is a robust

default—often numerically equal or slightly better than Cosine ,while Jaccard similarity is occasionally worse on titles and, more generally, more sensitive to token normalization.

Varying the size of reviewer profiles (number of prior publications) reveals a mild recency–coverage trade-off: restricting to recent work can improve early accuracy in fast-moving subfields while slightly reducing depth elsewhere. Removing venue-/year-specific tokens has negligible downside; acronym expansion and depluralization consistently help Jaccard similarity without harming Cosine, improving the *Cosine-Jaccard Average similarity* score. Ablating adjudication in the agentic LLM pipeline increases variance and can degrade title-based matching by introducing generic or redundant phrases. Taken together, these results suggest that (i) statistical differences among strong configurations are small at current sample sizes, (ii) lightweight normalization can stabilize Jaccard and benefit Cosine-Jaccard Average similarities, and (iii) ontology-aware structure (CSO relation edges) provides a repeatable, if modest, edge without fairness penalties.

Chapter 7

Conclusion

This thesis addressed the reviewer assignment problem (RAP) at conference scale by studying how different content signals—classical lexical features, LLM-generated keywords, and knowledge-graph (KG) representations, shape paper–reviewer affinity and flow into fair, capacity-aware assignments. Using a real-world dataset of 663 submissions from 85 venues and 524 anonymized reviewer profiles, we evaluated methods under realistic constraints with MRR, MAP, and P@k, and analyzed workload equity across venues. The result is a set of interpretable, reproducible building blocks for reviewer matching that balance accuracy, transparency, and operational practicality.

7.1 Summary of Contributions and Findings

Data, tasks, and metrics. We curated a domain-grounded corpus and standardized evaluation recipes (MRR, MAP, P@k) for comparing representations and similarity functions under conference constraints. This supports reproducible affinity studies and assignment analyses beyond toy settings.

Representation matters (RQ1). Representation and similarity choices materially affect ranking quality, with different text fields favoring different pipelines. On titles, where lexical evidence is sparse, an agent-adjudicated LLM keywording pipeline paired with the Cosine–Jaccard average delivered the most balanced performance (MRR = 0.7157, P@3 = 0.3922, MAP = 0.2993), while a constrained dot-product baseline remained the depth leader (MAP = 0.4367). On summaries, cosine with mean aggregation under constraints yielded the best depth (MAP = 0.6689), whereas dot-product with maximum aggregation gave the strongest classical top-k precision (P@3 = 0.3125). Among LLMs, Qwen 2.5 achieved the highest overall MRR (0.7155) on summaries; the agent pipeline matched or improved early accuracy while trading some depth. On abstracts, typed, ontology-aware KGs (CSO relations) produced the strongest aggregate quality (Cosine–Jaccard average: MRR = 0.7857, P@3 = 0.3920, MAP = 0.2936), outperforming both content-only and

LLM-only variants; among LLMs, compact keywording favored Jaccard (e.g., GPT-4o-mini MRR = 0.7229).

Robustness under incomplete/noisy profiles (RQ2). Aggregation and constraints mitigated sparse or noisy reviewer evidence. Capacity limits consistently regularized top-k volatility and improved depth (e.g., on summaries, cosine-mean improved by +31.7% MRR and +37.4% MAP when constraints were enforced), while max-aggregation helped surface a reviewer's single strong antecedent when evidence was uneven.

Interpretability vs. accuracy (RQ3). Two interpretable routes emerged. First, LLM-generated keyword sets (and their agent-adjudicated variants) offered concise, human-readable rationales and strong early precision particularly valuable on titles and summaries. Second, KG profiles combining OpenIE, GLiNER, and CSO topics provided structured, auditable signals and led overall performance on abstracts. Across settings, the Cosine–Jaccard average proved a robust fusion that balanced lexical overlap with semantic proximity.

From affinities to assignments and fairness (RQ4). When propagated to capacity-constrained matching, the improved affinities yielded realistic reviewer loads and low-to-moderate inequality at the conference level (macro-average $\approx$ 2.02–2.27, pooled mean $\approx$ 2.31–3.00, Gini $\approx$ 0.225–0.249), with CSO-based KGs exhibiting the lowest inequality among structured representations. These results indicate that semantically enriched signals diffuse matches beyond exact lexical ties and support more balanced allocations.

7.2 Limitations

The study focuses on Semantic Web / Computer Science venues and English-language abstracts; generalization to other disciplines and multilingual settings warrants caution. Reviewer profiles depend on public bibliographic quality (missing abstracts, author disambiguation), and LLM outputs remain sensitive to prompting and drift. While we analyzed workload equity, a full treatment of conflicts of interest and strategic behavior is outside scope. These caveats motivate the future directions outlined below.

7.3 Future Directions

Promising extensions include cross-venue calibration of similarity scales, multilingual KG enrichment and keywording, and uncertainty-aware assignments that propagate

confidence through ranking and matching. Further, integrating transparent conflict-of-interest graphs, learning-to-rank with fairness constraints, and human-in-the-loop rationale vetting could strengthen reliability and adoption in program-committee workflows.

7.4 Closing Remarks

Taken together, the results show that carefully engineered content signals—classical lexical features, interpretable LLM keywords, and ontology-aware KGs—can substantially improve reviewer matching at conference scale while preserving auditability and fairness under operational constraints. By pairing robust similarity functions (with Cosine–Jaccard averaging as a strong default) with realistic assignment objectives, this work contributes a transparent and scalable foundation for reviewer assignment systems that are not only effective, but also usable and trustworthy in real-world scholarly venues.

Bibliography

[1] A Ayodele Adebiyi, Olawole M Ogunleye, Marion Adebiyi, and JO Okesola. A comparative analysis of tf-idf, lsi and lda in semantic information retrieval approach for paper-reviewer assignment. *Journal of Engineering and Applied Sciences*, 14(10):3378–3382, 2019.

[2] Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H Laradji, Krishnamurthy Dj Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. Litllms, llms for literature review: Are we there yet? *arXiv preprint arXiv:2412.15249*, 2024.

[3] Meltem Aksoy, Seda Yanik, and Mehmet Fatih Amasyali. Reviewer assignment problem: A systematic review of the literature. *Journal of Artificial Intelligence Research*, 76:761–827, 2023.

[4] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI magazine*, 35(4):105–120, 2014.

[5] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, 2015.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[8] Lutz Bornmann. Scientific peer review. *Annual Review of Information Science and Technology*, 45:197–245, 2011.

[9] Chris Buckley and Ellen M Voorhees. Evaluating evaluation measure stability. In *ACM SIGIR Forum*, volume 51, pages 235–242. ACM New York, NY, USA, 2017.

[10] Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429. IEEE, 2023.

[11] Pedro G Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1):67–119, 2014.

[12] Lidia Ceriani and Paolo Verme. The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. *The Journal of Economic Inequality*, 10(3):421–443, 2012.

[13] Laurent Charlin and Richard S. Zemel. The toronto paper matching system: An automated paper–reviewer assignment system. In *ICML Workshop on Peer Reviewing and Publishing Models (ICML PeerReview)*, May 2013.

[14] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*, 2020.

[15] Victor Coscrato and Derek Bridge. Estimating and evaluating the uncertainty of rating predictions and top-n recommendations in recommender systems. *ACM Transactions on Recommender Systems*, 1(2):1–34, 2023.

[16] Cyrus Cousins, Justin Payan, and Yair Zick. Into the unknown: Assigning reviewers to papers with uncertain affinities. In *International Symposium on Algorithmic Game Theory*, pages 179–197. Springer, 2023.

[17] Nick Craswell. *Mean Reciprocal Rank*, pages 1703–1703. Springer US, Boston, MA, 2009.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, 2019.

[19] Davide Di Pierro, Eleonora Bernasconi, and Stefano Ferilli. Automatic reviews' assignments through answer set programming. 2025.

[20] Zhen Duan, Shicheng Tan, Shu Zhao, Qianqian Wang, Jie Chen, and Yanping Zhang. Reviewer assignment based on sentence pair modeling. *Neurocomputing*, 366:97–108, 2019.

[21] Thorsten Eisenhofer, Erwin Quiring, Jonas Möller, Doreen Riepel, Thorsten Holz, and Konrad Rieck. No more reviewer# 2: Subverting automatic {Paper-Reviewer} assignment using adversarial learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5109–5126, 2023.

[22] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[23] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[24] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[25] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[26] Stefano Ferilli, Nicola Di Mauro, Teresa Maria Altomare Basile, Floriana Esposito, and Marenglen Biba. Automatic topics identification for reviewer assignment. In *International conference on industrial, engineering and other applications of applied intelligent systems*, pages 721–730. Springer, 2006.

[27] Tanner Fiez, Nihar Shah, and Lillian Ratliff. A super* algorithm to optimize paper bidding in peer review. In *Conference on uncertainty in artificial intelligence*, pages 580–589. PMLR, 2020.

[28] Xian Gao, Jiacheng Ruan, Zongyun Zhang, Jingsheng Gao, Ting Liu, and Yuzhuo Fu. Reviewagents: Bridging the gap between human and ai-generated paper reviews. *arXiv preprint arXiv:2503.08506*, 2025.

[29] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[30] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[31] Jhih-Yi Janet Hsieh, Aditi Raghunathan, and Nihar B Shah. Vulnerability of {Text-Matching} in {ML/AI} conference reviewer assignments to collusions. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 5189–5208, 2025.

[32] Chao-Chun Hsu, Erin Bransom, Jenna Sparks, Bailey Kuehl, Chenhao Tan, David Wadden, Lucy Lu Wang, and Aakanksha Naik. Chime: Llm-assisted hierarchical organization of scientific studies for literature review support. *arXiv preprint arXiv:2407.16148*, 2024.

[33] Rahul Radhakrishnan Iyer, Manish Sharma, and Vijaya Saradhi. A correspondence analysis framework for author-conference recommendations. *arXiv preprint arXiv:2001.02669*, 2020.

[34] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura, 1901.

[35] Steven Jecmen, Nihar B Shah, Fei Fang, and Vincent Conitzer. Tradeoffs in preventing manipulation in paper bidding for reviewer assignment. *arXiv preprint arXiv:2207.11315*, 2022.

[36] Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. A survey on recent advances in named entity recognition. *arXiv preprint arXiv:2401.10825*, 2024.

[37] Firoz Khan, Mohammad Al Rawajbeh, Lakshmana Kumar Ramasamy, and Sangsoon Lim. Context-aware and click session-based graph pattern mining with recommendations for smart ems through ai. *IEEE Access*, 2023.

[38] Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1247–1257, 2019.

[39] Christin Katharina Kreutz and Ralf Schenkel. Revaside: assignment of suitable reviewer sets for publications from fixed candidate pools. In *The 23rd International Conference on Information Integration and Web Intelligence*, pages 57–68, 2021.

[40] Changlun Li, Yao Shi, Yuyu Luo, and Nan Tang. Rise of the community champions: From reviewer crunch to community power. *arXiv preprint arXiv:2503.18336*, 2025.

[41] Kaixia Li, Zhao Cao, and Dacheng Qu. Fair reviewer assignment considering academic social network. In *Web and Big Data: First International Joint Conference, APWeb-WAIM 2017, Beijing, China, July 7–9, 2017, Proceedings, Part I 1*, pages 362–376. Springer, 2017.

[42] Zhi-Qiang Li, Hui-Lin Xu, Hui-Juan Cao, Zhao-Lan Liu, Yu-Tong Fei, and Jian-Ping Liu. Use of artificial intelligence in peer review among top 100 medical journals. *JAMA Network Open*, 7(12):e2448609–e2448609, 2024.

[43] Xiang Liu, Torsten Suel, and Nasir Memon. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 25–32, 2014.

[44] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[45] Reza Yousefi Maragheh, Chenhao Fang, Charan Chand Irugu, Parth Parikh, Jason Cho, Jianpeng Xu, Saranyan Sukumar, Malay Patel, Evren Korpeoglu, Sushant Kumar, et al. Llm-take: Theme-aware keyword extraction using large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4318–4324. IEEE, 2023.

[46] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*, 2017.

[47] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[48] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[49] Kanika Mittal, Amita Jain, and Kunwar Singh Vaisla. Understanding reviewer assignment problem and its issues and challenges. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–6. IEEE, 2019.

[50] Kanika Mittal, Amita Jain, Kunwar Singh Vaisla, and Devendra K Tayal. A novel method for reviewer assignment problem based on reviewers' profile and fuzzy graph connectivity measure. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, pages 386–391. IEEE, 2020.

[51] Igor Nunes, Mike Heddes, Pere Vergés, Danny Abraham, Alex Veidenbaum, Alex Nicolau, and Tony Givargis. Dothash: Estimating set similarity metrics for link prediction and document deduplication. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1758–1769, 2023.

[52] Francesco Osborne and Enrico Motta. Klink-2: integrating multiple web sources to generate semantic topic networks. In *International Semantic Web Conference*, pages 408–424. Springer, 2015.

[53] Francesco Osborne, Angelo Salatino, Aliaksandr Birukou, and Enrico Motta. Automatic classification of springer nature proceedings with smart topic miner. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 383–399. Springer, 2016.

[54] Hongwei Peng, Haojie Hu, Keqiang Wang, and Xiaoling Wang. Time-aware and topic-based reviewer assignment. In *Database Systems for Advanced Applications: DASFAA 2017 International Workshops: BDMS, BDQM, SeCoP, and DMMOOC, Suzhou, China, March 27-30, 2017, Proceedings 22*, pages 145–157. Springer, 2017.

[55] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *Journal of machine learning research*, 22(164):1–20, 2021.

[56] Tribikram Pradhan, Suchit Sahoo, Utkarsh Singh, and Sukomal Pal. A proactive decision support system for reviewer recommendation in academia. *Expert Systems with Applications*, 169:114331, 2021.

[57] Sriraghavendra Ramaswamy. An efficient domain-independent approach for supervised keyphrase extraction and ranking. *arXiv preprint arXiv:2404.07954*, 2024.

[58] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[59] Dietrich Rordorf, Josua Käser, Alfredo Crego, and Emanuele Laurenzi. A hybrid intelligent approach combining machine learning and a knowledge graph to support academic journal publishers addressing the reviewer assignment problem (rap). In *AAAI Spring Symposium: MAKE*, 2023.

[60] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20, 2010.

[61] Angelo Salatino, Francesco Osborne, and Enrico Motta. Cso classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics. *International Journal on Digital Libraries*, 23(1):91–110, 2022.

[62] Angelo A Salatino, Andrea Mannocci, and Francesco Osborne. Detection, analysis, and prediction of research topics with scientific knowledge graphs. In *Predicting the dynamics of research impact*, pages 225–252. Springer, 2021.

[63] Angelo A Salatino, Francesco Osborne, Aliaksandr Birukou, and Enrico Motta. Improving editorial workflow and metadata quality at springer nature. In *International Semantic Web Conference*, pages 507–525. Springer, 2019.

[64] Angelo A Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. The cso classifier: Ontology-driven detection of research topics in scholarly articles. In *Digital Libraries for Open Knowledge: 23rd International Conference on Theory and Practice of Digital Libraries, TPDL 2019, Oslo, Norway, September 9-12, 2019, Proceedings 23*, pages 296–311. Springer, 2019.

[65] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[66] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[67] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[68] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2010.

[69] Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. Scirepeval: A multi-format benchmark for scientific document representations. *arXiv preprint arXiv:2211.13308*, 2022.

[70] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[71] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[72] Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. Is cosine-similarity of embeddings really about similarity? *arXiv preprint arXiv:2403.05440*, 2024.

[73] Ivan Stelmakh, Nihar B Shah, and Aarti Singh. Peerreview4all: Fair and accurate reviewer assignment in peer review. In *Algorithmic Learning Theory*, pages 828–856. PMLR, 2019.

[74] Ivan Stelmakh, John Wieting, Graham Neubig, and Nihar B Shah. A gold standard dataset for the reviewer assignment problem. *arXiv preprint arXiv:2303.16750*, 2023.

[75] Vaios Stergiopoulos, Michael Vassilakopoulos, Eleni Tousidou, Spyridon Kavvathas, and Antonio Corral. Conference management system utilizing an llm-based recommendation system for the reviewer assignment problem. 2025.

[76] Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. *arXiv preprint arXiv:2504.09737*, 2025.

[77] João Vinagre, Alípio Mário Jorge, and João Gama. Fast incremental matrix factorization for recommendation with positive-only feedback. In *International conference on user modeling, adaptation, and personalization*, pages 459–470. Springer, 2014.

[78] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

[79] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[80] Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, et al. Scimmir: Benchmarking scientific multi-modal information retrieval. *arXiv preprint arXiv:2401.13478*, 2024.

[81] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357, 2024.

[82] Yaoguang Yong, Zheng Yao, and Yawei Zhao. A framework for reviewer recommendation based on knowledge graph and rules matching. In *2021 IEEE international conference on information communication and software engineering (ICICSE)*, pages 199–203. IEEE, 2021.

[83] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*, 2023.

[84] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long*

Papers), pages 5364–5376, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[85] Peng Zhang, Peng Fu, Kaixuan Chen, and Jing Zhang. A novel paper-reviewer recommendation method based on a semantics and correlation fusion model. In *Proceedings of the International Conference on Computing, Machine Learning and Data Science*, pages 1–6, 2024.

[86] Tianmai M Zhang and Neil F Abernethy. Reviewing scientific papers for critical problems with reasoning llms: Baseline approaches and automatic evaluation. *arXiv preprint arXiv:2505.23824*, 2025.

[87] Tong Zhang, Yinan Zhang, Min Xin, Jiashe Liao, and Qingfeng Xie. A lightweight network for small insulator and defect detection using uav imaging based on improved yolov5. *Sensors*, 23(11):5249, 2023.

[88] Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

[89] Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. Pre-training multi-task contrastive learning models for scientific literature understanding. *arXiv preprint arXiv:2305.14232*, 2023.

[90] Xiquan Zhao and Yangsen Zhang. Reviewer assignment algorithms for peer review automation: A survey. *Information Processing & Management*, 59(5):103028, 2022.

[91] Ruiyang Zhou, Lu Chen, and Kai 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)*, pages 9340–9351, 2024.