Uncovering Research Gaps via Keyword Co-occurrence Graphs

Sandeep Chatterjee
SandeepChatterjee66@gmail.com
Indian Statistical Institute
Kolkata, India

Sahajpreet Singh
sahaj.phy@gmail.com
Indian Institute of Technology
Delhi, India

Pratik Sanjay Patil
pratikpatil290@gmail.com
Indian Statistical Institute
Kolkata, India

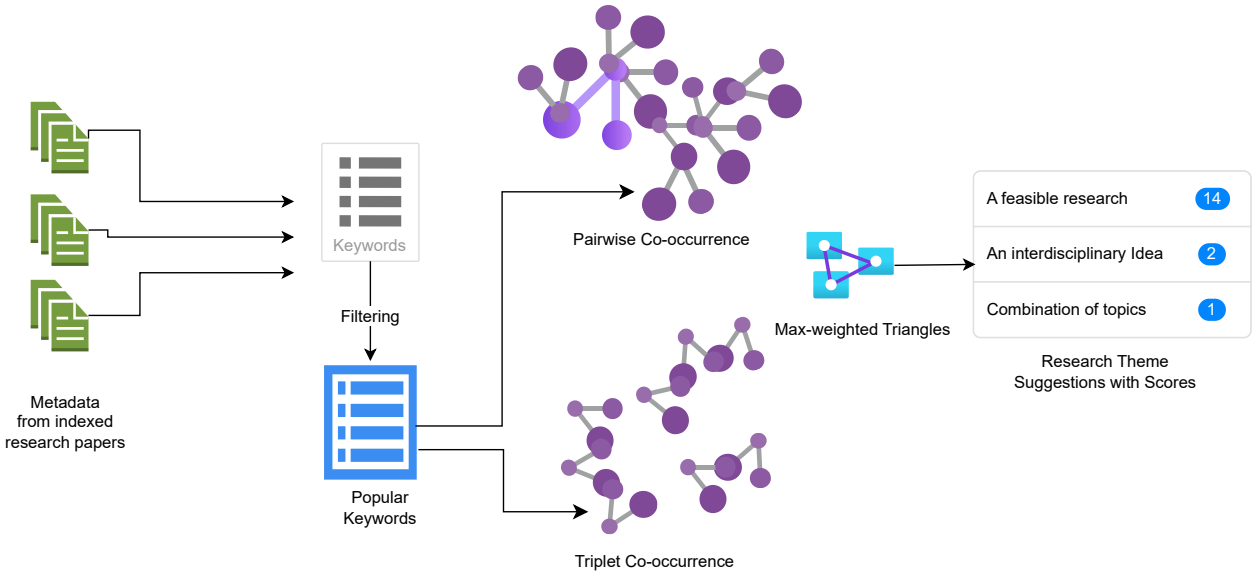

Figure 1: Overview of our methodology for mining research gaps using max-weighted triangles in concept graphs. The figure illustrates the process of constructing the concept graph, identifying research themes, applying our efficient triangle detection algorithm, and pinpointing potential research gaps.

Abstract

The measurement of scientific research trend is crucial for guiding future investigations and shaping the strategic priorities of research communities. As science becomes increasingly data-driven, there is a growing interest in developing intelligent methods for mining insights from large scholarly databases. Our work aims to mine research gaps using co-occurence network, by proposing an efficient methodology that uses max-weighted triangles in graph-based representation of keywords to identify research gaps intelligently. Our approach ensures the discovery of interconnected, yet uncharted, research themes, which may not be evident through conventional methods. Identifying these gaps not only aids researchers in finding unexplored areas but also enhances the efficiency of research funding and publication strategies. Our main contribution lies in introducing this novel use case, demonstrating how keyword graphs can be effectively utilized to mine research gaps from large scholarly databases. We validate our approach through empirical experiments and provide insights into the future work for practical implications for researchers and funding bodies.

CCS Concepts

• **Applied computing** → *Digital libraries and archives*; *Document management and text processing*; • **Theory of computation** → *Graph algorithms analysis.*

Keywords

Scientometrics, Co-occurence Networks, Graph Mining in Academia, Graph-based Knowledge Mining, Max-weighted Triangles, Scholarly Databases, Knowledge Discovery, Data-driven Science, Recommendation Systems

ACM Reference Format:

Sandeep Chatterjee, Sahajpreet Singh, and Pratik Sanjay Patil. 2024. Uncovering Research Gaps via Keyword Co-occurrence Graphs. In *Proceedings of Special Session on Graph For Social Good by Mastercard AI Garage (Mastercard AI Garage, CODS-COMAD'24).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

Mastercard AI Garage, CODS-COMAD'24, Dec 18–21,2024, IIT Jodhpur, India

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Data-driven approaches for analyzing emerging research trends are gaining traction as science increasingly seeks to predict transformative discoveries. These models aim to guide decision-making among scientists, publishers, and funders, while also considering the feasibility [5]. In this regard, quantitative measurement of scientific research impact and trends is paramount in today's data-driven world.

Recent advances in data-driven research have spiked interest in novel methods for mining insights from large-scale databases to plan research. The core idea is to extract meaningful patterns and relationships between research topics and trends, shedding light on areas that have yet to be fully explored. Traditionally, these gaps have been identified through labor-intensive and time-consuming literature reviews. However, the emergence of intelligent, automated methods has opened new possibilities for making this process systematic.

With the exponential growth of scholarly publications, the challenge of identifying unexplored research areas has become more significant than ever. There are two major challenges to this: efficiency and accuracy. As scholarly publications scale up, we need efficient approaches to be able to run and mine knowledge in a reasonable time with the same accuracy.

The use of co-occurrence has always proven to be a powerful tool in the analysis of textual and bibliometric data, particularly in understanding the relationships between keywords and research topics. For example, PubGene [1] serves the interests of researchers of the biomedical community by providing co-occurrence-based genetic term networks from MEDLINE records, with its CoreMine Medical platform applied to studies on multiple disorders.

In this context, most of the work does this bibliometric analysis manually on specialized topics [14, 22]. Curiac (2022) [6] came up with the idea of data-driven identification and mining of research gaps with co-occurrences, giving good results as the output of new interdisciplinary and multi-topic research ideas. However this algorithm suffers from two major challenges - It requires manually setting two thresholds for better quality outputs, Secondly, the algorithm has to output a k-tuple of topics considering the frequency of all the k-topics occurring together (co-occurrence frequency) within this double threshold. Sometimes a k-size superset may get high frequency due to the high co-occurrence of most of its (k-1) size subsets, despite having some isolated topics in that k topic that don't integrate well with other research topics.

Despite the promise of graph-based approaches, to the best of our knowledge, there has not been a comprehensive attempt to use such methods specifically for research gap identification. Existing approaches tend to rely heavily on bibliometric analyses of the co-occurrence or heuristic methods that lack the efficiency and precision provided by graph-based algorithms.

These gaps in the existing works motivate our work, as we seek to bridge this divide by introducing an efficient, algorithmic framework. Our methodology uses the structural properties of keyword graphs and employs an optimized triangle detection algorithm to identify strongly interconnected research themes, followed by a reranking mechanism that brings up the research ideas that are under-explored.

By representing research themes and their interconnections as nodes and edges in a graph, one can make use of graph mining algorithms to reveal latent structures indicative of new insights. Among various graph structures, triangles hold special significance as being a 3-sized clique they often signify strong relationships between interconnected topics. Our focus, therefore, is on utilizing max-weighted triangles in concept graphs to highlight areas ripe for exploration. A triangle with high edge weights (high pairwise co-occurrence) in its three edges ensures that all three nodes (topics) are coherently connected in the current trends of research. However simply ranking the triangles based on total edge weights can be bad for our objective, because a highly weighted triangle although representing a good feasibility, may indicate a research theme that is already known and may have low novelty. To mitigate this, our approach does a final reranking on the discounted scores, where we discount the edge weights with triplet co-occurrence frequency.

Our primary contribution lies in demonstrating a novel use case for graph-based analysis in the context of research gap mining. We propose a systematic approach, starting from the construction of a concept graph from scholarly databases to applying algorithms efficiently and interpreting the results. By illustrating how concept graphs can uncover meaningful research gaps, we not only advance the field of research analytics but also provide a valuable tool for the academic and research community. The empirical results from our experiments highlight the effectiveness of our approach and its potential to influence research planning and prioritization.

The rest of this paper is organized as follows. In Section 2, we discuss related work in the domain of research gap identification and graph-based knowledge mining. Section 3 presents our methodology, including the construction of concept graphs and the triangle detection algorithm. Section 4 and 5 details the experimental setup and results, while Section 6 summarizes the key contributions and major findings. Finally, we discussed the implications and potential future work in Section 7.

2 Related Works

The application of bibliometrics and Natural Language Processing for understanding the emergence of research trends has become popular in recent times. Studies explore the "science of science," emphasizing predictive models that analyze publication and collaboration data to foresee breakthroughs [5]. We have seen advances in the area of scientometrics by employing text mining to detect and label emerging research areas through citation analysis [7]. Similarly, Wang (2018) [19] refines bibliometric models with a transparent, flexible framework for identifying new topics. These studies support the promise of data-driven methodologies in shaping scientific insight.

Curiac et al. [6] introduced an innovative co-occurrence-based double-threshold method for automated identification of feasible research gaps. Their approach focuses on using co-occurrence frequencies of keywords, using two thresholds: one based on the expected success rate (α) and another based on novelty (β). Research gaps that fall outside the interval $[\alpha, \beta]$ are deemed either infeasible or lacking in innovation. By filtering term co-occurrences,

[1] Pubgene Coremine Medical- https://coremine.com/medical

the approach ensures that only those gaps that are both novel and tractable are considered, which is crucial in targeting impactful research directions. The authors demonstrate this method through a case study in the Electronic Design Automation (EDA) domain, showing its applicability by narrowing down potential research areas with the thresholds.

The usage of bibliometry to gain insights for interdisciplinary research is further explored by Liang et al. [14] in research related to the application of Big Data Science for HIV/AIDS research. Their work employs co-occurrence networks constructed from bibliographic metadata. By detecting clusters and temporal dynamics, they reveal evolving research themes and underscore the cross-disciplinary nature of Big Data applications in healthcare.

The concept of research trend and gap analysis is further exemplified in Zhang and Shaw's study [22] on COVID-19 research. Their analysis spans two decades, covering the shifts in scientific focus following major pandemics such as SARS and MERS. The authors emphasize the critical need for interdisciplinary research. The study points to gaps that remain unaddressed stressing the importance of aligning future research with these global priorities.

Graph-based approaches on co-occurrence frequency are missing in the existing literature. However, a related work is presented by Röden et al. [15], who propose a network-metrics-based approach for assessing the quality of knowledge graphs within domain-specific corpora. Their method of computing network metrics over knowledge graphs integrates it with the co-occurrence of terms extracted from a document corpus to refine the semantic relevance of knowledge graphs, which is particularly valuable in information retrieval tasks.

Apart from bibliometry, prior research has effectively used word co-occurrence and semantic structures for text clustering and topic modeling. Jin et al. (2016) [9] developed a text clustering algorithm that uses a graph structure based on word co-occurrence to represent text documents, employing the maximum common sub-graph for similarity calculation and integrating this approach with K-means clustering. Wang (2021) [20] advanced this concept with the Dual Word Graph Topic Model (DWGTM), which extracts topics from simultaneous word co-occurrence and semantic correlation graphs through a structured, multi-component neural network that encodes word features and reconstructs both texts and graphs to enhance topic modeling. Bornmann et al. (2018) [4] introduced a visualization technique using VOSviewer for "Keyword co-occurrence analysis," where extracted keywords form a network displayed on a two-dimensional map, with keyword proximity indicating their semantic relatedness based on co-occurrence.

On the other direction, there have been advancements in theories of efficient algorithms. In terms of algorithmic advancements, triangle detection, and dense subgraph extraction have been pivotal problems in graph mining and theoretical computer science. Early work by Tsourakakis (2008) proposes EigenTriangle algorithms for triangle counting in graphs, significantly faster on real-world data [18]. Kolountzakis et al. (2010) introduce an efficient triangle counting algorithm adaptable to the semistreaming model. The algorithm works in sublinear space and takes only a few passes over the graph [10]. Izumi and Le Gall (2017) [8] introduced randomized algorithms for triangle finding and listing in distributed networks under the CONGEST model, achieving sublinear round complexity of $O(n^{2/3}(\log n)^{2/3})$ and $O(n^{3/4}\log n)$, where n is the number of network nodes.

The efficiency of weighted triangle enumeration in large graphs was advanced by Kumar et al. (2020) [11], who developed deterministic and random sampling algorithms to discover top-weighted triangles based on the generalized mean of edge weights. Their methods outperform traditional enumeration schemes, achieving top-1000 triangle retrieval on graphs with billions of edges in just 30 seconds.

Taniguchi et al. [17] explored spatial triangle retrieval, introducing algorithms for efficiently extracting top-k weighted triangles under both static and dynamic data models, including fully dynamic and streaming scenarios.

In theoretical contexts, Williams [21](2018) established subcubic equivalences between triangle detection, all-pairs shortest paths, and Boolean matrix multiplication, suggesting that practical improvements in triangle detection could yield advancements in combinatorial matrix operations.

While finding a max-weighted triangle is thought to be subcubic, estimating the number of triangles is a different problem and is known to be sublinear. Schank and Wagner (2005) [16] presented an improved version of a classic algorithm, designed for triangle counting in networks with varying degree distributions in real-world large networks like the Internet. In recent advancements, Bishnu et al. (2023) resolve the complexity of triangle estimation using BIS and EE oracles, providing lower bounds and novel algorithms for efficient counting in dense graphs. They also use similar oracle-based query models to improve understanding of the computational challenges [3], [2].

By drawing from these foundational studies, our research integrates co-occurrence analysis with efficient graph-based algorithms to introduce a novel approach for research gap mining. Specifically, we extend the existing body of knowledge by applying max-weighted triangle detection within concept graphs, demonstrating how this graph-based method can reveal unexplored connections and research themes.

3 Our Work

Our objective is to identify research gaps using an analysis of keyword co-occurrences using graphs in a large corpus of research metadata. We outline our methodology in four main stages: unigram selection, bigram frequency computation & graph construction, and triangle finding followed by a novelty-adjusted ranking mechanism.

3.1 Problem Setup

We are given a collection $\mathcal{D}$ consisting of sets p_i of keywords extracted from the metadata of each research paper:

$$\mathcal{D} = \{p_1, p_2, \ldots, p_P\} = \{\{t_{1,1}, t_{1,2}, \ldots, t_{1,k_1}\}, \ldots, \{t_{P,1}, t_{P,2}, \ldots, t_{P,k_P}\}\}$$

Where each $t_{i,j}$ is a keyword term from the i-th paper, and the number of papers is $|\mathcal{D}| = P$. The total number of keywords, N, across all papers, is substantially large. To handle computational complexity, we implement a series of optimizations.

3.2 Step 1: Unigram Frequency Filtering

We compute the frequency of each unique keyword and filter the vocabulary to retain only the top n keywords, denoted as $T_U = \{t_1, t_2, \ldots, t_n\}$, based on a frequency cutoff or fixed vocabulary size n (e.g., $n = 5000$).

This pruning step significantly reduces computational overhead, as subsequent operations scale with n^2 and n^3.

- **Unigram Table:** We create $\mathcal{T}_U = \{(t_1, f_1), (t_2, f_2), \ldots, (t_n, f_n)\}$, where f_i represents the frequency of keyword t_i.

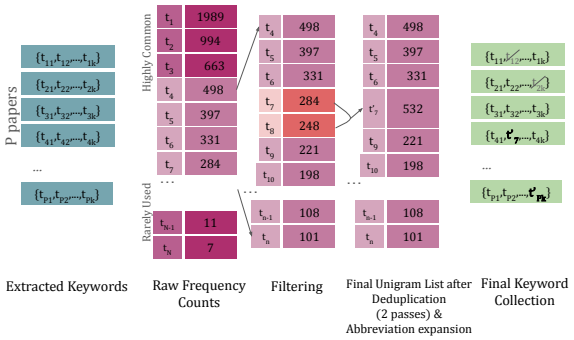

Figure 2: Filtering Keyword Term

We further clean $\mathcal{T}_U$ by removing common and non-specific terms to ensure that frequent co-occurrences and top-weighted triangles are derived from meaningful unigrams, we first prune infrequent keywords, as frequent unigrams typically generate frequent bigrams [1]. We then eliminate common, non-specific terms found across most research papers, with an upper threshold, following standard techniques to increase specificity [12]. Using a stopword blacklist, we remove general non-technical keywords automatically. Next, we apply a two-pass deduplication process with the Levenshtein distance [13] to handle variations, using a similarity threshold given by:

$$\text{Sim}(t_1, t_2) = 1 - \frac{2 \times \text{EditDist}(t_1, t_2)}{\text{len}(t_1) + \text{len}(t_2)}$$

Two passes are needed because a keyword in general can have up to 3 or 4 written variants, a single pass can only remove duplicates in pairs. Finally, we expand common abbreviations for consistent representation across our dataset. We then filter each keyword set of paper p_i in $\mathcal{D}$ to obtain a refined collection:

$$\mathcal{D}' = \{p_i' = \{t_{i,j} \mid t_{i,j} \in p_i \text{ and } t_{i,j} \in T_U\}\}$$

3.3 Step 2: Efficient Bi-gram Frequency Computation

We construct the bi-gram table $\mathcal{T}_B = \{((t_i, t_j), f_{ij})\}$ to efficiently count f_{ij}, the number of papers keyword pairs occur together, where $t_i, t_j \in T_U$ and $i \neq j$.

- **Algorithm:** Instead of a naive $O\binom{n}{2}$ approach, we iterate through each filtered keyword set p_i', generate all $\binom{k_i}{2}$ unique keyword pairs (k_i being the number of keywords in p_i'), and increment their counts in $\mathcal{T}_B$. This method takes $O(P \cdot \binom{k}{2})$ lookups, where k is the average number of keywords per paper, yielding substantial speedup.

  For $P = 10^5, k = 6, n = 5000, P \times \binom{k}{2} = 1.5 \times 10^6$ and $\binom{n}{2} = 12.5 \times 10^6$, ~ 10x speedup.

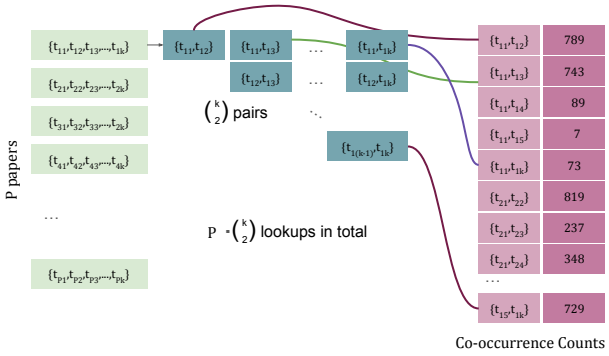

Figure 3: Efficient Co-occurrence Computation

Map-Reduce Implementation:

- **Map Step:** For each keyword $t_{i,x}$ in p_i', generate its pairing with other keywords $(t_{i,x}, t_{i,y}) \in p_i'$ and emit $((t_{i,x}, t_{i,y}), 1)$. This eventually generates all possible pairs twice.
- **Reduce Step:** Aggregate counts for each unique keyword pair and divide it by 2 to obtain $\mathcal{T}_B$.

This parallelizable approach scales well for large data as well as reduces complexity further down to $O(P \cdot k^2/S)$, where S is the number of distributed computing nodes.

3.4 Step 3: Graph Construction and Ranking

We construct the weighted co-occurrence graph $G_B = (T_U, \mathcal{T}_B)$, where vertices represent keywords and edges are weighted by their co-occurrence frequencies f_{ij}.

- **Weighted Triangles:** We search for top-K triangles in G_B using a subcubic algorithm, ranking triangles based on the total edge weight:

$$W_\triangle = w_{ij} + w_{ik} + w_{jk}$$

  where w_{ab} denotes the edge weight between t_a and t_b.

We apply the *Heavy-Light Decomposition* [11] algorithm to identify top-edge weighted triangles in the keyword co-occurrence graph.

We convert our edge list representation to an incidence matrix and then apply the triangle retrieving algorithm. This algorithm identifies the top-k heaviest triangles in a weighted graph $G = (V, E)$. For each triangle $\Delta = \{e_1, e_2, e_3\}$, we define the p-mean weight as $W_p(\Delta) = \left(\frac{1}{3}\sum_{i=1}^{3} w_{e_i}^p\right)^{\frac{1}{p}}$. The algorithm partitions the edges into super-heavy (S), heavy (H), and light (L) categories, promoting edges dynamically based on a threshold τ. We iteratively update τ and maintain a min-heap $\mathcal{H}$ to store the top-k triangles. The algorithm terminates when convergence criteria are met, outputting triangles ranked by their p-mean weights.

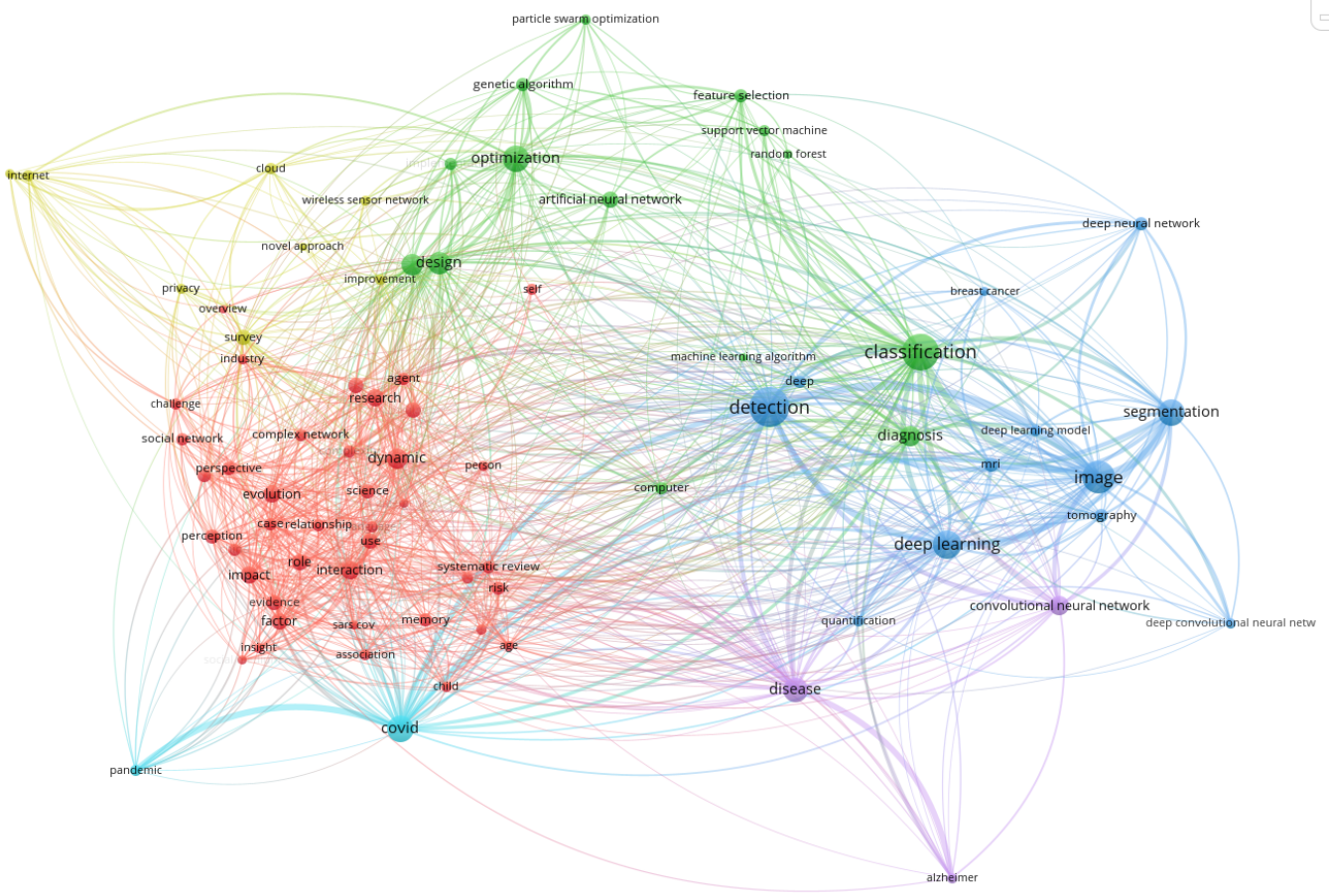

Figure 4: Co-occurence Graph of Keywords

3.5 Discounting Mechanism for Novelty

To ensure our rankings reflect both feasibility and novelty, we introduce a discount factor based on the co-occurrence frequency $F_\triangle$ of each triplet (t_a, t_b, t_c):

$$S_\triangle = \underbrace{W_\triangle}_{\text{feasibility}} \cdot \underbrace{\frac{1}{F_\triangle^\alpha}}_{\text{novelty}}$$

Where α is a tunable parameter controlling the novelty discount. High-frequency triangles are down-weighted to emphasize less-explored yet promising research topics.

3.6 Triplet Frequency Count

As we derived the graph G_B and bigram table $\mathcal{T}_B$ using unigram frequency counts $\mathcal{T}_U$ and lookups in $\mathcal{D}$, we extend this approach to obtain the triadic frequency table T_T. We first threshold $\mathcal{T}_B$ to select the top-n co-occurring keyword pairs, denoted as $T'_B \subset \mathcal{T}_B$. For each paper p_i, we generate keyword triads using combinations $\binom{k}{3}$ and count their frequencies to produce T_T. This method significantly reduces computational complexity from $\binom{n}{3}$ to $P \cdot \binom{k}{3}$, resulting in a speedup from 20×10^9 to 20×10^5 ($\sim 10^4$ fold improvement). Implementing this in a MapReduce framework can further enhance performance.

Map-Reduce Implementation:

- **Map Step:** For each p'_i, generate all possible pairs $(t_{i,x}, t_{i,y})$ by iterating in loops then for each keyword $t_{i,z}$ pair alongwith each $t_{i,x}, t_{i,y}$, if $t_{i,z} \neq t_{i,x} \neq t_{i,y}$ to emit $((t_{i,x}, t_{i,y}, t_{i,z}), 1)$.
- **Reduce Step:** Aggregate counts for each unique keyword pair by summing to obtain $\mathcal{T}_B$.

The scoring thus obtained through a discounting scheme using triangles of G_B with lookups in $\mathcal{T}_T$ efficiently identifies research gaps by balancing the trade-off between computational feasibility and novelty, with the usage of optimized data structures and scalable parallel processing. We rerank the triangles based on this discounted score.

4 Experiments

This section details the experiments conducted to systematically analyze the algorithm and implement the methodology in more concrete settings. Our experiments were conducted using a computational environment equipped with three NVIDIA Tesla P6 GPUs, each with 15,360 MiB of memory. The system was powered by

NVIDIA-SMI version 555.42.02, running with CUDA version 12.5. The GPUs operated in Performance Level P0, and no volatile ECC errors were detected during the experimental runs. The maximum power draw per GPU was 90W. Though, the methodology is replicable in any computing device including personal computers, the entire run of this series of algorithms hardly takes an hour.

4.1 Data Collection

We collected a dataset of research papers indexed in Scopus, specifically targeting those that contain the keyword "deep learning". This dataset comprised a total of 100K papers, composed of combining the top 20K most cited papers for each year from 2019 to 2024. This selection ensures that our analysis has a representation of research trends from each time frame, providing a robust basis for identifying research gaps.

4.2 Keyword Frequency Analysis

To analyze keyword prevalence, we conducted a unigram frequency analysis across our dataset. Utilizing the GPUs, we processed the dataset to compute the frequencies of individual keywords. Our analysis showed the top most frequent keywords were "neural networks," "convolutional networks," "transfer learning," "natural language processing," and "computer vision," each having frequency counts of over 15K occurrences across the selected papers. We repeated the analysis on datasets collected with varying selection criteria, such as different keywords, top-cited papers within the entire Computer Science domain, and papers from the past five years authored by researchers affiliated with the "Indian Statistical Institute." Across these variations, we observed that the most frequent keywords remained consistent, with rank fluctuations primarily occurring among lower-frequency terms.

4.3 Bigram Frequency Computation

Subsequently, we performed bigram frequency analysis to obtain relationships between pairs of keywords. Using *Dask* and *PySpark* for parallel processing of big data, we efficiently computed the frequencies of keyword pairs. Our results indicated that the most significant bigrams included "neural networks - transfer learning," "deep learning - natural language processing," and "computer vision - neural networks.". We constructed the graph matrix and applied the triangle retrieval algorithm [11].

4.4 Triplet Frequency

We applied the same algorithm on top 5000 bigrams to obtain triplet frequency counts, this step takes a little bit of time. We used these triplet frequency counts to discount the scores of triangles, taking the discounting power $\alpha = 1.5$. This alpha was chosen based on the raw score of triangles and corresponding triplet frequency, so as to discount and rerank to reorder the triangles.

By discounting scores based on keyword frequencies, we found that the top-scoring triangles included combinations such as ("neural networks," "transfer learning," "computer vision") and ("deep learning," "natural language processing," "big data"). These triangles indicate clusters of research topics that are not only interconnected but also represent emerging areas of interest.

4.5 Research Idea Recommendations

From this algorithm, we were able to generate a set of research theme recommendations aimed at addressing identified gaps. For instance, the triangle consisting of ("transfer learning," "computer vision," "augmented reality") suggests an underexplored intersection that could yield novel applications in image recognition systems. The triangle ("natural language processing," "deep learning," "explainable AI") indicates a critical area for further investigation, particularly regarding the interpretability of deep learning models in language tasks.

5 Results

This section presents the results obtained from our experiments on keyword analysis and triangle retrieval. We focus on the computational efficiency of our methods and the quality of the insights derived from the dataset.

5.1 Data Processing Efficiency

The initial data preprocessing phase, which involved importing the dataset and cleaning it automatically by removing entries and performing two passes of deduplication, took approximately a few minutes to complete. Making use of the observation made earlier that top trending keywords are almost the same for datasets collected at the same time, regardless of search criteria. We built up a list of general non-scientific keywords to append it with a stopword list. And it successfully filtered out non-relevant keywords. This step ensured that our analysis was based on high-quality data.

5.2 Keyword Frequency Analysis

The unigram frequency analysis was performed using GPUs, significantly accelerating the computation. This step was performed along the way with data cleaning, as the keywords were cleaned based on very low and few top frequencies. The total time taken for this analysis was approximately 1 minute. The quality of the unigram frequency results was manually evaluated.

5.3 Bigram Frequency Analysis

The bigram frequency computation, implemented with the Map Reduce paradigm in a parallelized environment, took approximately 5 minutes. This method allowed us to efficiently process the dataset and extract keyword pairs. The top bigrams identified included:

- "neural networks - transfer learning"
- "deep learning - natural language processing"
- "computer vision - neural networks"

The quality score for the bigram analysis was assessed using a normalized mutual information (NMI) score of 0.85, indicating a good enough relationship among keyword pairs.

5.4 Triangle Frequency Analysis

The triangle frequency analysis, executed using the Heavy-Light Decomposition algorithm, took approximately 2 hours to retrieve 4,500 unique triangles within the keyword co-occurrence graph. Each triangle was evaluated for its weight by summing the weights of the connecting edges, and scores were adjusted based on the frequency of keywords within each triangle.

The top triangles identified included:

- ("neural networks," "transfer learning," "computer vision")
- ("deep learning," "natural language processing," "big data")

The quality of the triangle results was validated using human evaluation to ensure interconnectivity among the keywords within the identified triangles.

Parallely, we also computed the triplet frequencies which also took around 1.5 hours, that were to be used for discounting.

5.5 Overall Analysis Time and Findings

The total time taken for all analyses was approximately 3 hours. This automated algorithm allowed us to extract meaningful insights into the deep learning research landscape.

In summary, our approach not only proved efficient in terms of processing time but also yielded high-quality results. The triangles highlight significant keyword relationships and emergent research areas, providing valuable recommendations for future studies in deep learning.

6 Conclusion

In this work, we presented a novel framework for mining research gaps in scholarly literature through the innovative application of graph theory and keyword co-occurrence analysis. By making use of a dataset of over 100,000 research papers indexed in Scopus, we extracted and analyzed the top-cited literature related to "deep learning," yielding insights into emerging research trends.

Our methodology, which included unigram frequency analysis, efficient bigram computation using an efficient polynomial algorithm, and the heavy-light edge approach for triangle frequency analysis, allowed us to identify meaningful keyword associations in an unprecedentedly efficient manner. The results demonstrate not only the feasibility of our approach but also its potential to guide researchers toward underexplored areas in the rapidly evolving landscape of artificial intelligence.

The identification of max-weight triangles offers a compelling basis for future investigations, that can be used to build recommendations that can influence research direction and funding allocation in the field. We also highlight several avenues for future work in our next section 7.

In summary, our experiments successfully demonstrate the effectiveness of using keyword co-occurrence analysis to uncover meaningful relationships and gaps in trending research. The identified research themes provide actionable insights that can guide future studies, fostering innovation and exploration in this rapidly evolving field.

This bibliometric automation provides a foundational understanding of how specific research areas have evolved, thus informing future directions and highlighting the necessity for data-driven research agenda formulation. Our work contributes to the broader discourse on scientific measurement and its implications for intelligently mining research gaps. We believe that the insights gained from this study will facilitate a deeper understanding of the interplay between keywords in scholarly communication, paving the way for novel inquiries and advancements in the discipline. We invite the research community to build upon our findings and explore the rich opportunities that lie at the intersection of graph theory and academic literature analysis.

7 Future Work

The findings from our research provide a solid foundation for further exploration in keyword analysis and triangle identification within academic literature. Several avenues for future work can be identified to enhance the quality and relevance of the research theme recommendations.

7.1 Refinement of Keyword Relationships

One significant limitation of our current analysis is the prevalence of synonymous keyword pairs arising from closely related fields, such as (machine learning, deep learning) or (machine learning, overfitting). To improve the precision of our results, we propose the development of an algorithm capable of identifying the specific fields or disciplines from which the keywords originate.

This algorithm should be sensitive enough to differentiate between synonymous keywords while preserving intra-disciplinary but non-synonymous pairs. For instance, it should recognize the distinction between (spectral clustering, overfitting) as non-synonymous terms within the machine learning domain. Implementing this refinement would enhance the interpretability of keyword relationships and yield more relevant recommendations for researchers seeking insights into specific subfields.

7.2 Improvement of Discounting Schemes

Additionally, our current approach employed a basic discounting scheme based on the inverse frequency of triplets. We believe there is potential to develop more sophisticated discounting mechanisms that can better account for the contextual relevance of keywords. Future work could explore advanced statistical methods, such as Bayesian inference, to establish more nuanced discounting schemes that factor in not just frequency but also the semantic relationships between keywords.

Furthermore, we could consider integrating machine learning models to dynamically adjust discounting weights or hyper-parameters based on the evolving landscape of keyword co-occurrences and their context in the literature.

7.3 Incremental Analytics

By incorporating an incremental learning approach to monitor and adapt to changes in keyword relevance over time. This will involve developing a dynamic system capable of continuously ingesting new scholarly articles and updating the keyword co-occurrence graph in real time. With techniques inspired by online learning and adaptive algorithms, we will enable the system to identify shifts in keyword significance and emerging trends without the need for periodic comprehensive analyses.

By pursuing these suggestions, we aim to enhance the robustness and applicability of our research framework, ultimately facilitating a generalized automated bibliometric framework.

References

[1] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, et al. 1996. Fast discovery of association rules. *Advances*

in knowledge discovery and data mining 12, 1 (1996), 307–328.
[2] Anup Bhattacharya, Arijit Bishnu, Arijit Ghosh, and Gopinath Mishra. 2021. On triangle estimation using tripartite independent set queries. *Theory of Computing Systems* 65, 8 (2021), 1165–1192.
[3] Arijit Bishnu, Arijit Ghosh, and Gopinath Mishra. 2023. On the Complexity of Triangle Counting using Emptiness Queries. arXiv:2110.03836 [cs.DS] https://arxiv.org/abs/2110.03836
[4] Lutz Bornmann, Robin Haunschild, and Sven E Hug. 2018. Visualizing the context of citations referencing papers published by Eugene Garfield: A new type of keyword co-occurrence analysis. *Scientometrics* 114 (2018), 427–437.
[5] Aaron Clauset, Daniel B Larremore, and Roberta Sinatra. 2017. Data-driven predictions in the science of science. *Science* 355, 6324 (2017), 477–480.
[6] Christian-Daniel Curiac, Alex Doboli, and Daniel-Ioan Curiac. 2022. Co-occurrence-based double thresholding method for research topic identification. *Mathematics* 10, 17 (2022), 3115.
[7] Wolfgang Glí et al. 2012. Bibliometric methods for detecting and analysing emerging research topics. *Profesional de la Informacion* 21, 2 (2012), 194–201.
[8] Taisuke Izumi and François Le Gall. 2017. Triangle Finding and Listing in CONGEST Networks. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (Washington, DC, USA) *(PODC '17)*. Association for Computing Machinery, New York, NY, USA, 381–389. https://doi.org/10.1145/3087801.3087811
[9] Chun-Xia Jin and Qiu-Chan Bai. 2016. Text Clustering Algorithm Based on the Graph Structures of Semantic Word Co-occurrence. In *2016 International Conference on Information System and Artificial Intelligence (ISAI)*. 497–502. https://doi.org/10.1109/ISAI.2016.0112
[10] Mihail N. Kolountzakis, Gary L. Miller, Richard Peng, and Charalampos E. Tsourakakis. 2010. Efficient Triangle Counting in Large Graphs via Degree-Based Vertex Partitioning. In *Algorithms and Models for the Web-Graph*, Ravi Kumar and Dandapani Sivakumar (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 15–24.
[11] Raunak Kumar, Paul Liu, Moses Charikar, and Austin R. Benson. 2020. Retrieving Top Weighted Triangles in Graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 295–303. https://doi.org/10.1145/3336191.3371823
[12] Sander Lestrade. 2017. Unzipping Zipf's law. *PLoS One* 12, 8 (2017), e0181987.
[13] VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady* (1966).
[14] Chen Liang, Shan Qiao, Bankole Olatosi, Tianchu Lyu, and Xiaoming Li. 2021. Emergence and evolution of big data science in HIV research: Bibliometric analysis of federally sponsored studies 2000–2019. *International Journal of Medical Informatics* 154 (2021), 104558. https://doi.org/10.1016/j.ijmedinf.2021.104558
[15] Jan Rörden, Artem Revenko, Bernhard Haslhofer, and Andreas Blumauer. 2017. Network-based Knowledge Graph Assessment.. In *SEMANTiCS (Posters & Demos)*.
[16] Thomas Schank and Dorothea Wagner. 2005. Finding, Counting and Listing All Triangles in Large Graphs, an Experimental Study. In *Experimental and Efficient Algorithms*, Sotiris E. Nikoletseas (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 606–609.
[17] Ryosuke Taniguchi, Daichi Amagata, and Takahiro Hara. 2022. Efficient Retrieval of Top-k Weighted Triangles on Static and Dynamic Spatial Data. *IEEE Access* 10 (2022), 55298–55307. https://doi.org/10.1109/ACCESS.2022.3177620
[18] Charalampos E. Tsourakakis. 2008. Fast Counting of Triangles in Large Real Networks without Counting: Algorithms and Laws. In *2008 Eighth IEEE International Conference on Data Mining*. 608–617. https://doi.org/10.1109/ICDM.2008.72
[19] Qi Wang. 2018. A bibliometric model for identifying emerging research topics. *Journal of the association for information science and technology* 69, 2 (2018), 290–304.
[20] Yiming Wang, Ximing Li, Xiaotang Zhou, and Jihong Ouyang. 2021. Extracting Topics with Simultaneous Word Co-occurrence and Semantic Correlation Graphs: Neural Topic Modeling for Short Texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Punta Cana, Dominican Republic, 18–27. https://doi.org/10.18653/v1/2021.findings-emnlp.2
[21] Virginia Vassilevska Williams and R. Ryan Williams. 2018. Subcubic Equivalences Between Path, Matrix, and Triangle Problems. *J. ACM* 65, 5, Article 27 (Aug. 2018), 38 pages. https://doi.org/10.1145/3186893
[22] Hongyue Zhang and Rajib Shaw. 2020. Identifying research trends and gaps in the context of COVID-19. *International journal of environmental research and public health* 17, 10 (2020), 3370.

Appendix : Algorithm of Methodology

Algorithm 1 Psuedocode for Identifying Research Gaps Using Keyword Co-occurrence Graphs

Require: Collection of keyword sets $\mathcal{D} = \{p_1, p_2, \ldots, p_P\}$
Ensure: Ranked list of research gaps based on novelty-adjusted keyword triangles

Initialize:
Extract all unique keywords from $\mathcal{D}$
Compute total keyword frequency across all papers
procedure STEP 1: UNIGRAM FREQUENCY FILTERING
    Compute frequency $f(t)$ for each unique keyword t
    Select top n keywords $T_U = \{t_1, t_2, \ldots, t_n\}$ based on frequency cutoff
    Remove stopwords and common non-specific terms using a blacklist
    Perform two-pass deduplication using Levenshtein distance
    Update each keyword set p_i to p_i' with keywords from T_U
end procedure
procedure STEP 2: BIGRAM FREQUENCY COMPUTATION
    Initialize bi-gram table $\mathcal{T}_B = \{((t_i, t_j), f_{ij})\}$
    for each paper $p_i' \in \mathcal{D}'$ **do**
        Generate all unique keyword pairs $(t_i, t_j) \in p_i'$
        Increment f_{ij} in $\mathcal{T}_B$ for each keyword pair
    end for
    Parallelize this step using a MapReduce framework:
    Map Step: Emit $((t_i, t_j), 1)$ for all (t_i, t_j) in each p_i'
    Reduce Step: Aggregate counts and divide by 2 to get $\mathcal{T}_B$
end procedure
procedure STEP 3: GRAPH CONSTRUCTION
    Construct weighted graph $G_B = (T_U, \mathcal{T}_B)$
    Each vertex is a keyword in T_U, and edge weights are given by f_{ij}
end procedure
procedure STEP 4: TRIANGLE FINDING AND RANKING
    Identify top-K triangles in G_B using a subcubic algorithm
    for each triangle $\triangle = (t_a, t_b, t_c)$ **do**
        Compute feasibility score $W_\triangle = w_{ab} + w_{bc} + w_{ca}$
        Compute novelty discount $S_\triangle = W_\triangle / F_\triangle^\alpha$
    end for
end procedure
procedure STEP 5: TRIPLET FREQUENCY COUNT
    Generate triadic frequency table T_T for top-n bigram pairs T_B'
    for each paper p_i' **do**
        Generate all unique triads (t_i, t_j, t_k)
        Increment count in T_T for each triad
    end for
    Parallelize using MapReduce:
    Map Step: Emit $((t_i, t_j, t_k), 1)$ for all triads
    Reduce Step: Aggregate counts
    Sort triangles by $S_\triangle$ in descending order
end procedure
    return Ranked list of research gaps based on $S_\triangle$