S&DS 625 CapstoneReport: Analysis and Clustering of Academic Papers Using LDA and Network Analysis

Ruixiao Wang

December 15, 2023

Contents

1 Abstract

This project involves an in-depth analysis of academic papers from the Association for Computing Machinery (ACM) journals. Using community detection algorithms and Latent Dirichlet Allocation (LDA), it uncovers collaborative networks, identifies trending research topics, and spots influential authors within the ACM community. The study focuses on large research communities, employing network analysis to discover key figures and themes in computing related research. It aims to assist newcomers in identifying relevant research groups and connecting with established experts in their areas of interest, fostering academic growth and collaboration.

2 Executive Summary

This project delves into analyzing a vast dataset from the Association for Computing Machinery (ACM) journals, focusing on the collaborator network, trending research topics, active researchers, and influential authors. Utilizing community detection algorithms and Latent Dirichlet Allocation (LDA), the project identifies patterns of collaboration and knowledge exchange, discerns dominant themes, and pinpoints influential authors. The dataset includes over 629,000 papers and 632,000 citation relationships. The methodology involves community detection, topic modeling, and influential author identification. Results reveal key research clusters, prevalent topics, and notable researchers, offering a comprehensive understanding of ACM's academic landscape. This analysis is vital for researchers, especially newcomers, in identifying relevant research groups and influential figures, enhancing academic collaboration and growth.

3 Introduction

In this project, our objective was to conduct an in-depth analysis of a substantial dataset comprising academic papers published in the Association for Computing Machinery (ACM) journals. ACM, a leading international academic society, focuses on advancing computing as a science and profession. Its journals are renowned for featuring cutting-edge research in the field of computing and information technology. Our analysis particularly centers on unraveling the collaborator network within these publications, identifying trending research topics, investigating active researchers, and examining the influential authors within the ACM community.

To achieve these goals, we utilized community detection algorithms and Latent Dirichlet Allocation (LDA). Community detection helped in identifying prevalent groups or clusters of authors within the network, highlighting patterns of collaboration and knowledge exchange. LDA was applied to discern dominant themes and topics across the corpus of papers, shedding light on the evolving landscape of research interests. Additionally, we employed network analysis techniques to pinpoint influential authors, thereby gaining insights into key figures driving research trends and collaborations.

4 Dataset

The dataset for this analysis was sourced from https://www.aminer.org/citation, in particular the one named Citation-network V1, comprising 629,814 papers along with more than 632,752 citation relationships, providing a comprehensive view of academic contributions and influences as of May 15, 2010. This rich dataset served as a foundation for our exploration into the intricate dynamics of academic research within the ACM community.

5 Preliminaries

This section provides an overview of the fundamental concepts and methodologies utilized in the analysis, including graph theory, community detection, topic modeling, similarity measurement, and influential author identification.

5.1 Graph Theory Basics

Graph theory is a field of mathematics and computer science concerned with the properties of graphs. A graph G is defined as a set of nodes (or vertices) V and a set of edges E that connect pairs of nodes. Formally, $G = (V, E)$. In the context of this project, nodes represent authors, and edges represent co-authorship relationships between them.

5.2 Latent Dirichlet Allocation (LDA) Topic Modeling

Latent Dirichlet Allocation (LDA) is a generative probabilistic model used for uncovering the underlying thematic structure in large collections of documents. LDA posits that each document can be described as a mixture of various topics, and each topic, in turn, is characterized by a distribution over words.

5.2.1 LDA Algorithm Outline

: The LDA model assumes the following generative process for each document in a corpus:

1. Choose a distribution over topics, $\theta_d \sim Dirichlet(\alpha)$. Here, θ_d represents the topic distribution for document d, and α is the parameter of the Dirichlet prior on the per-document topic distributions.

2. For each word $w_{d,n}$ in document d:

   (a) Choose a topic $z_{d,n} \sim Multinomial(\theta_d)$.

(b) Choose a word $w_{d,n}$ from $p(w_{d,n}|z_{d,n}, \beta)$, a multinomial probability conditioned on the topic $z_{d,n}$. The parameter β represents the Dirichlet prior on the per-topic word distribution.

Inference and Parameter Estimation: Given the observed documents, the goal is to infer the hidden topic structure. This is achieved by computing the posterior distribution of the hidden variables given a document:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)}$$

Since exact inference is intractable for LDA, approximation methods such as Variational Bayes or Markov Chain Monte Carlo (MCMC) sampling are typically used.

5.3 Similarity Measurement

5.3.1 Levenshtein Distance in Fuzzy String Matching

The FuzzyWuzzy library utilizes the concept of Levenshtein Distance to define the similarity between two strings. The Levenshtein Distance is a string metric for measuring the difference between two sequences. It is calculated as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other.

Levenshtein Distance: Given two strings, a and b, with lengths $|a|$ and $|b|$, respectively, the Levenshtein Distance $D(a, b)$ is given by $D(a_{|a|}, b_{|b|})$, where $D(i, j)$ is recursively defined as:

$$D(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0, \\ \min \begin{cases} D(i-1,j)+1, \\ D(i,j-1)+1, \\ D(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases}$$

Here, $1_{(a_i \neq b_j)}$ is the indicator function, equal to 0 when $a_i = b_j$ and equal to 1 otherwise. This recursive definition computes the distance based on the previous computations, building up the solution from smaller subproblems.

Similarity Score: FuzzyWuzzy uses this distance to calculate a similarity score, which is a normalized measure, expressed as a percentage. The score is calculated as:

$$SimilarityScore = \left(1 - \frac{LevenshteinDistance}{\max(|a|, |b|)}\right) \times 100$$

This score reflects how similar two strings are: a score of 100 indicates a perfect match, while a lower score indicates less similarity. FuzzyWuzzy uses this score to identify the closest matches for a given string from a collection of strings, enabling effective fuzzy matching even when there are small typos, variations in spelling, or other minor differences.

5.4 Cosine Similarity

In this project, similarity measurement between words is used to find clusters relevant to a given set of input words. Cosine similarity is a common measure, defined as:

$$cosinesimilarity = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \times ||\mathbf{B}||}$$

where $\mathbf{A}$ and $\mathbf{B}$ are two non-zero vectors.

6 Data Exploration and Visualization

6.1 Characteristics of Collaboration and Citation Networks

In the process of exploring the dataset, two types of networks were constructed and analyzed: the collaboration (coauthor) network and the citation network. Each network offers distinct insights into the academic landscape.

Collaboration (Coauthor) Network:

- **Number of Nodes (Authors)**: The collaboration network comprises 505,027 nodes, each representing an individual author.
- **Number of Edges (Coauthorships)**: There are 1,137,185 edges in the network, with each edge signifying a coauthorship relation between two authors.
- This network provides a comprehensive view of the collaborative relationships within the academic community, highlighting patterns of coauthorship and group dynamics in research activities.

Citation Network:

- **Number of Nodes (Papers)**: The citation network contains 629,814 nodes, corresponding to individual academic papers.
- **Number of Edges (Citations)**: There are 632,751 edges, each indicating a citation from one paper to another.
- However, a significant portion of the raw data consisted of empty references. Specifically, out of the total papers, only 125,372 had non-empty reference lists, suggesting potential missing data or incomplete citation information.

6.2 Prioritizing Collaboration Network Analysis

Given the data characteristics and the observed incompleteness in the citation network, the decision was made to prioritize the analysis of the collaboration network. The reasons for this prioritization include:

- **Data Completeness**: The collaboration network exhibited a higher degree of data completeness and reliability compared to the citation network, which contained a substantial number of empty references.
- **Insightful Metrics**: The collaboration network allows for the examination of coauthorship patterns, identification of research clusters, and influential author analysis, providing valuable insights into active research collaborations and community structures.
- **Impact on Research Dynamics**: Analyzing the collaboration network offers a direct perspective on the dynamics of academic research collaboration, which is fundamental to understanding research productivity and influence in the field.

Therefore, the subsequent analysis focused predominantly on the collaboration network, leveraging its rich and more complete dataset to extract meaningful patterns and insights into the academic research landscape.

6.3 Collaboration Network Subgraph Visualization

After constructing the collaboration network, I selected a couple of renowned professors in the field of statistics to visualize their networks, focusing specifically on their direct neighbors. The visualization is academically meaningful as it reveals the nuances of student-advisor relationships, collegial collaborations, and shared research interests, etc, as shown in Figure 1 and Figure 2.

6.4 Partitioning into Large and Small Communities

Initial Findings from Community Detection: After applying community detection algorithms to the authorship network of ACM publications, approximately 50,000 distinct clusters were identified. However, upon closer inspection, it was observed that only 19 clusters had a substantial number of papers (over 15,000 in total). This disparity in clusters sizes led to a strategic decision in the analysis process: partitioning the communities into two categories, namely 'large' and 'small' communities.
Criteria for Partitioning: The criterion for this partitioning was based on the total number of papers within each community. Communities with more than 5,000 papers were classified as 'large', while those with fewer were deemed 'small'. This partitioning was predicated on the hypothesis that larger communities, owing to their substantial academic output, are likely to be more active and collaborative.
Focus on Large Communities: The decision to focus subsequent analysis on large communities was driven by several considerations:

- **Active and Collaborative Authorship**: Larger communities are likely to represent more active researchers and a higher degree of collaboration, potentially leading to more influential and groundbreaking research.
- **Influence and Popularity of Research Topics**: Topics prevalent in larger communities are presumed to be more representative of the dominant trends and interests in the field at the time of the dataset's compilation.

The analysis of large communities thus aimed to provide a deeper understanding of the core academic contributions and collaborative patterns in the ACM publications, offering valuable insights into the landscape of computing research during the period covered by the dataset.

7 Methodology

To summarize, I first apply **Community Detection** on the Collaboration Network and identify large clusters (communities). For the large communities, a detailed analysis was conducted, involving:

- **Topic Modeling**: Applying LDA to extract prevalent topics within these communities.
- **Influential Author Identification**: Using network centrality measures to identify and analyze the most influential authors within these communities.
- **Visualization and Interpretation**: Creating visual representations of the network and topic distributions to aid in the interpretation of findings.

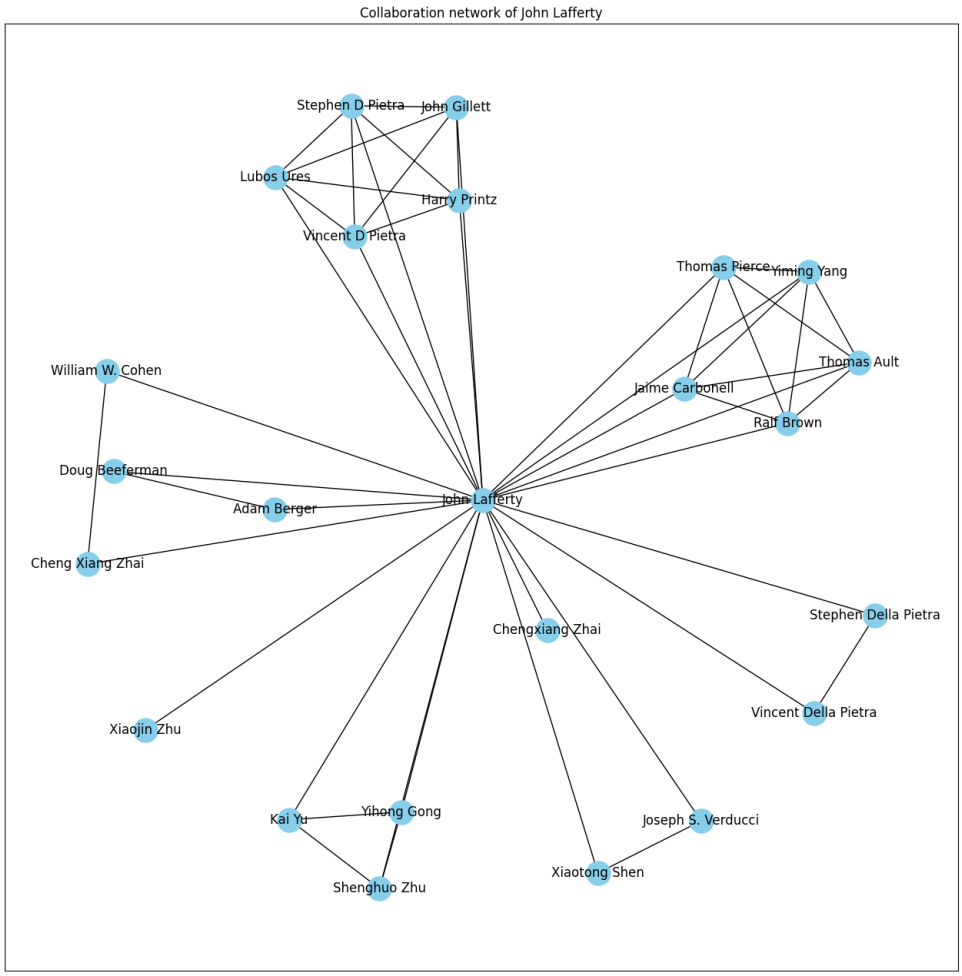

Figure 1: Visualization of Professor John Lafferty's Collaborative Network. The subgraph highlights significant connections in Professor Lafferty's academic collaborations. For example, it includes Chengxiang Zhai, who was one of his Ph.D. students. The network also features Dr. William W. Cohen, who held various positions at Carnegie Mellon University's Machine Learning groups and intersected professionally with Professor Lafferty when they were both at CMU, sharing similar research interests. Additionally, Stephen Della Pietra, associated with the IBM Thomas J. Watson Research Center, represents a connection from a period when Professor Lafferty was also affiliated with IBM Research. This subgraph serves as a testament to the vibrant and influential research ecosystem to which Professor Lafferty contributes.

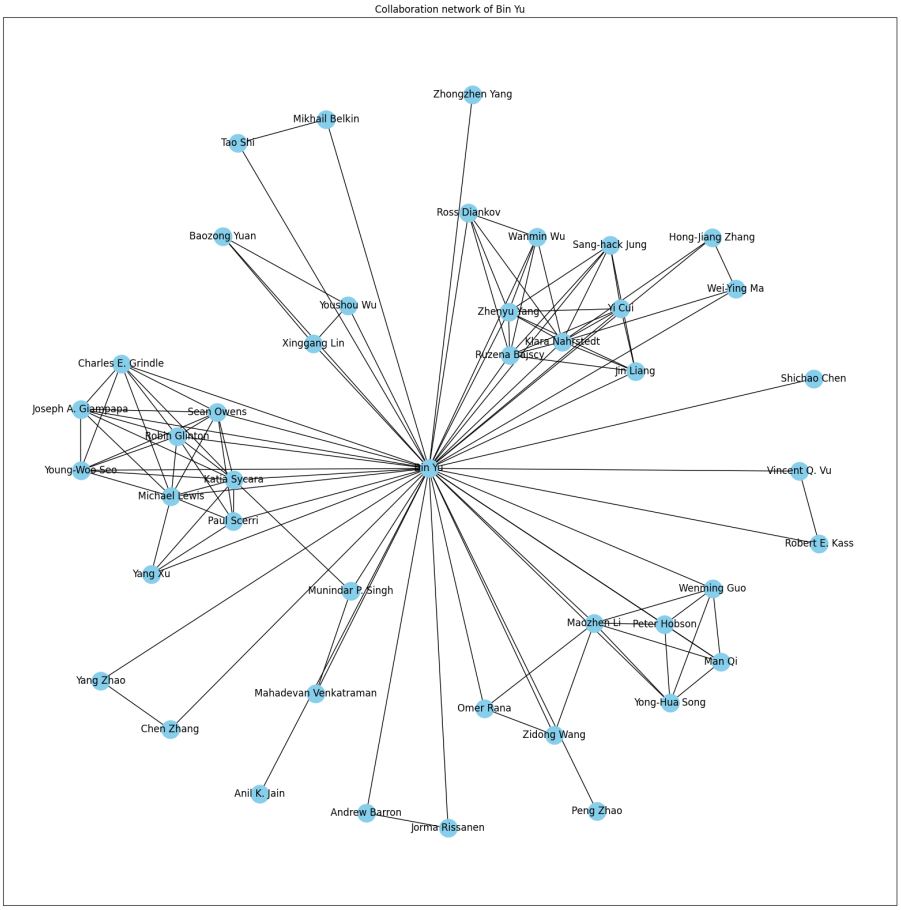

Figure 2: Visualization of Professor Bin Yu's Collaborative Network at the University of California, Berkeley. As an esteemed statistician, Professor Yu's network underscores her extensive academic collaborations. This subgraph not only maps her significant connections but also reflects the broader potential of leveraging collaboration networks within ACM to distill valuable insights into the dynamics of scholarly research.

7.1 Data Preprocessing

- **Data Collection**: Collected academic papers with details including titles, authors, abstracts, and citations.
- **Data Cleaning**: Implemented text preprocessing steps, including tokenization, stopword removal, and lemmatization, to prepare the data for LDA.

7.1.1 Matching Author Names Using Fuzzy String Matching

In the analysis of academic papers, accurately identifying authors is crucial, especially given the variations in name representations. To address this challenge, the project employed a fuzzy string matching technique to match author names, ensuring a higher degree of accuracy in identifying authors across the dataset.

The approach involved the following steps:

1. **Normalization**: All author names in the dataset were converted to lowercase to maintain uniformity and avoid mismatches due to case differences.
2. **Fuzzy Matching**: For each name to be matched, the fuzzy string matching algorithm, implemented via the *fuzzywuzzy* Python library, was used to find the closest match within the set of all author names. The FuzzyWuzzy library utilizes the concept of Levenshtein Distance to define the similarity between two strings.
3. **Top Match Selection**: If an exact match for a name was not found, the algorithm selected the top similar name based on a similarity score. This score quantifies how close a candidate name is to the input name, considering factors like letter arrangements and phonetic similarity.

Implementation: The implementation involved the use of the *fuzzywuzzy* library's *process* module. The function *find_similar_names* takes an input name, normalizes it to lowercase, and then searches for the most similar name within the set of all author names, also normalized to lowercase. If there is no exact match, the function returns the top similar name based on the calculated similarity score.

Rationale: This method is particularly effective in handling common issues in author name matching, such as typos, different name orderings, or variations in name spelling. By using fuzzy matching, the project could more accurately aggregate papers under the correct authors, thereby improving the reliability of subsequent analyses like author collaboration networks and influencer identification.

Limitations and Considerations: While fuzzy string matching enhances name matching accuracy, it is not infallible. It may sometimes incorrectly match names that are similar but represent different individuals. Thus, the results obtained through this method were used with careful consideration, particularly when drawing conclusions about individual authors' research contributions and collaborations.

7.2 Community Detection

Applied Louvain community detection algorithms to identify clusters of authors within the network.

7.2.1 Attributes and Advantages

The Louvain method for community detection is renowned for its efficiency and effectiveness, especially in large networks. Key attributes and advantages of the Louvain method as compared to other community detection algorithms include:

- **Scalability**: Efficiently handles large networks, making it ideal for datasets with thousands or millions of nodes.
- **Modularity Optimization**: The Louvain method optimizes modularity, a scale-independent measure, allowing for meaningful community detection in networks of varying sizes.
- **Hierarchical Clustering**: It reveals community structures at multiple scales, from small subgroups to larger clusters, by iteratively merging communities.
- **Speed and Simplicity**: The algorithm is faster than many other community detection methods, due to its heuristic approach based on modularity maximization.

In comparison, other methods like the Girvan-Newman algorithm or spectral clustering may offer different perspectives on community structure but often at a higher computational cost, particularly for very large networks.

7.2.2 Role of Community Detection in this Project

Community detection, particularly through the Louvain method, plays a pivotal role in this project by:

- **Identifying Research Clusters**: By detecting communities, the analysis can pinpoint clusters of authors who frequently collaborate or work on similar topics, indicating active research areas within ACM.
- **Enhancing Topic Analysis**: Understanding community structures enhances the interpretation of LDA results, as topics are often shared or prevalent within certain communities.
- **Guiding Network Analysis**: Community detection informs subsequent network analysis, including the identification of influential authors, by providing a clearer structure within the authorship network.

7.2.3 Algorithm of Louvain Method

The Louvain method is an algorithm for detecting communities in large networks based on modularity optimization. Here is an outline of the algorithm:

1. **Initialization**: Start with a network where each node is in its own community. Thus, initially, there are as many communities as there are nodes.
2. **Modularity Optimization**: For each node i, consider its neighboring nodes and the communities to which these neighbors belong. Move node i to the community that results in the largest gain in modularity. Repeat this process iteratively for all nodes until no further increase in modularity is possible.

3. **Community Aggregation**: Once the first pass is complete, create a new network. In this new network, each node represents a community from the previous step. The weight of the edge between two new nodes is equal to the sum of the weights of the edges between nodes in the corresponding two communities.

4. **Iterate**: Repeat steps 2 and 3 iteratively until modularity cannot be increased further. The result is a hierarchy of communities, with the final iteration yielding the community divisions of the network.

5. **Modularity Calculation**: The modularity Q at each stage is calculated to measure the strength of the community structure found. The algorithm seeks to maximize this value.

The modularity Q is given by the formula:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} is the weight of the edge between nodes i and j, k_i and k_j are the sum of the weights of the edges attached to nodes i and j, m is the sum of all of the edge weights in the graph, and $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same community and 0 otherwise.

The Louvain method is particularly efficient due to its hierarchical approach, quickly identifying community structures at various scales and allowing for the analysis of large networks.

7.3 LDA Topic Modeling

- **Model Building**: Built LDA models to extract topics from the academic papers. Tuned parameters such as the number of topics and words per topic.

- **Model Evaluation**: Evaluated models using coherence scores to ensure topic quality and interpretability.

Application in the Project: In this project, LDA was applied to identify the prevalent topics within the ACM publications. By analyzing the distribution of topics across documents, insights into the research focus and trends within the community were obtained. The coherence of the topics generated by LDA was also evaluated to ensure their interpretability and relevance to the research questions being addressed.

7.4 Influential Author Identification Techniques

Various centrality measures are used to identify influential authors in a network:
Degree Centrality: The degree centrality for a node v is the fraction of nodes it is connected to.

$$C_D(v) = \frac{deg(v)}{|V| - 1}$$

Betweenness Centrality: This measures the number of times a node acts as a bridge along the shortest path between two other nodes.

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)}$$

PageRank: PageRank computes a ranking of the nodes in the graph based on the structure of the incoming links.

The combined score used in the project is an average of these centrality measures for each author in the subgraph:

$$combined\_score(v) = \frac{C_D(v) + C_B(v) + PageRank(v)}{3}$$

These measures collectively provide a robust mechanism for identifying key players in the academic collaboration network.

7.5 Research Group Finding

Our approach utilizes Latent Dirichlet Allocation (LDA) for topic modeling, coupled with TF-IDF based cosine similarity computation. This method aims to efficiently identify research groups in each large clusters that focus on topics closely related to user-specified interests. In short, we compare user specified topic keywords and compare the similarity of these keywords and keywords in every classified topics within each large cluster, and provide the user with top 3 relevant topics in each cluster.

The implemented approach serves as an invaluable tool for researchers, particularly those embarking on new academic endeavors. It not only highlights relevant research topics but also facilitates networking with established experts, fostering academic growth and collaboration.

8 Results

8.1 Community Detection in Collaborator Network

The community detection analysis of the collaborator network yielded a total of 52,505 clusters. For the purpose of in-depth analysis and NLP topic modeling, clusters with a significant number of papers (over 5000) were particularly noted. The following table summarizes the number of papers associated with these major clusters:

This analysis reveals that while most clusters are relatively small, several clusters contain a substantial number of papers, indicating more active and extensive collaboration within these communities. Such clusters were prioritized for further natural language processing and topic analysis to extract meaningful insights into the prevalent research themes and patterns of collaboration within these larger communities.

8.2 Example: Cluster 30

Cluster 30 has 1735 papers in total, which is large enough for interesting analysis, and also small enough for visualization purpose. In the analysis of cluster 30 within the ACM dataset, the collaboration network of several renowned figures in computing and AI was visualized, revealing meaningful academic relationships. The top influential authors in this cluster include Manish Gupta, William J. Dally, Andrew A. Chien, Julian Dolby, and Philip Heidelberger. Their research interests align closely with specific topics identified through LDA topic modeling, highlighting common themes in high-performance computing, compiler optimization, and scalable systems. A visualization of cluster 30 is shown in Figure 3.

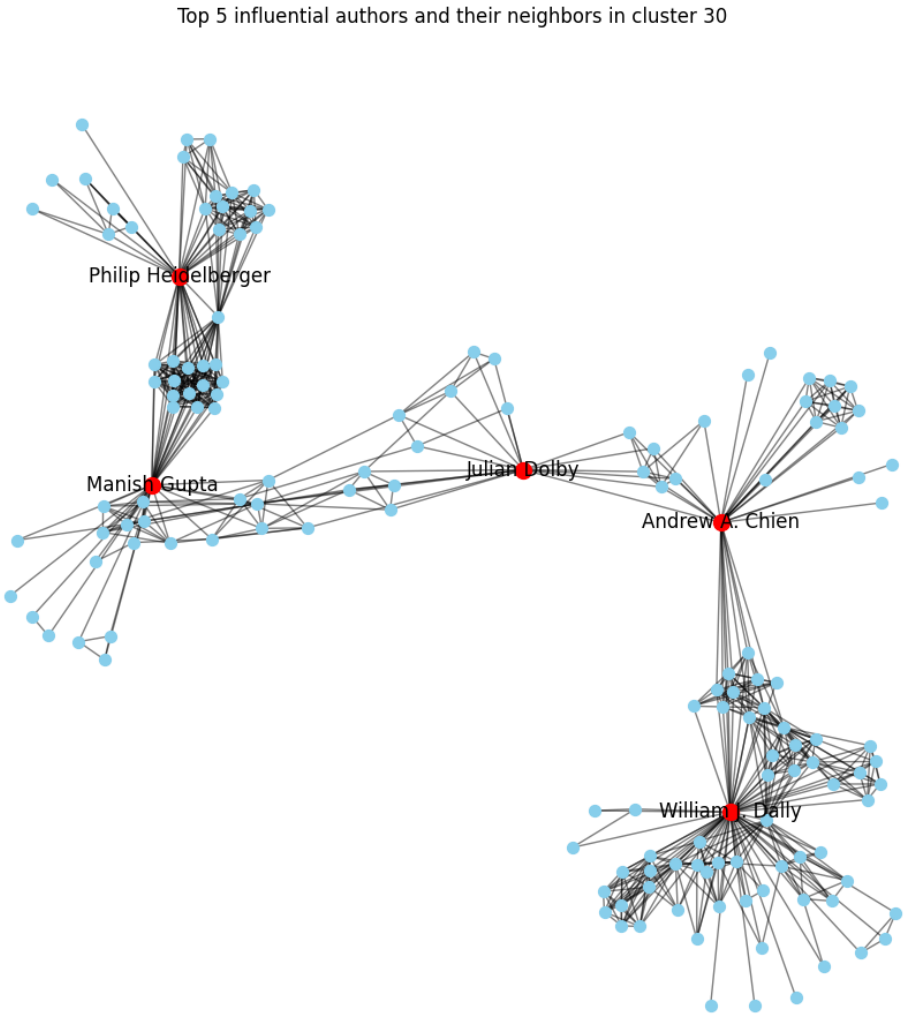

Figure 3: Cluster 30 visualization with top 5 influential authors highlighted in red.

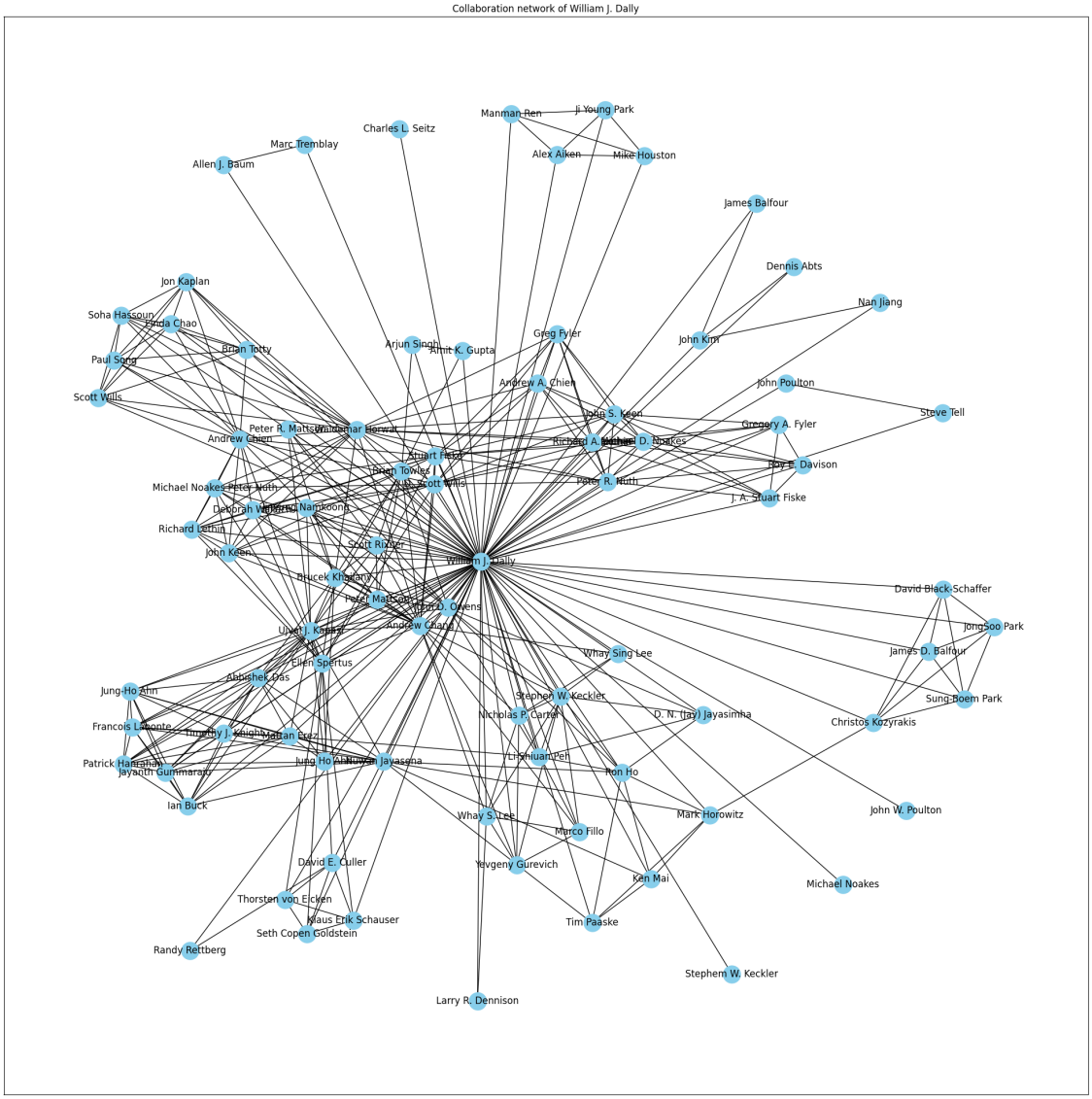

Figure 4: William J. Dally, top influential author in cluster 30, collaboration network visualization.

Cluster Number	Number of Papers
1	5493
4	18325
7	32098
9	7728
10	6921
25	7993
28	13710
29	5103
31	15936
43	5351
51	12773
62	9111
70	5732
72	13775
88	10244
100	5626
109	16350
120	10393
149	24060

Table 1: Major Clusters and Their Associated Number of Papers in the Collaborator Network

- Manish Gupta: His work in system software for supercomputers like Blue Gene/L aligns with Topic 3 (cache, chip, and latency) and Topic 7 (instruction, execution and compiler optimization). Gupta's focus on big data analytics also connects with Topic 12 (database and message processing).
- William J. Dally: As a specialist in computer architecture and parallel programming, Dally's research intersects with Topic 7 (compiler optimization and dynamic execution) and Topic 9 (adaptive routing and interconnection networks).
- Andrew A. Chien: His work on large-scale sustainable systems and cloud computing is reflected in Topic 12 (database and message processing) and Topic 18 (software and scheduling). Chien's interest in renewable energy and system architecture resonates with the environmental aspects of computing.
- Julian Dolby: Dolby's research in program analysis, semantic web, and AI is evident in Topic 4 (object-oriented programming and software development) and Topic 19 (object collection and user interaction). His work in scripting languages and security analysis connects with web-based technologies.
- Philip Heidelberger: Known for his focus on interconnection networks and computer architecture in supercomputing environments, his expertise is linked with Topic 3 (cache and chip access) and Topic 9 (routing and network topology).

A common thread among these authors is their significant contributions to the field of high-performance computing and scalable systems. Their collective work, encapsulating various aspects

Topic	Keywords
1	distributed, communication, node, software, cluster, protocol, component, resource, multiprocessor, output
2	compiler, optimization, image, branch, power, efficiency, error, communication, hardware, information
3	cache, chip, access, latency, bandwidth, compilation, different, pattern, server, hardware
4	object, type, oriented, programming, view, support, distributed, pointer, policy, distribution
5	protocol, state, interval, steady, confidence, framework, mean, multiple, consistency, process
6	computer, connection, interface, message, communication, verification, operating, programming, logic, passing
7	instruction, execution, scheduling, compiler, code, load, optimization, dynamic, heuristic, hardware
8	sampling, predictor, importance, procedure, estimation, measure, neural, carlo, monte, estimate
9	routing, adaptive, router, cost, cube, traffic, interconnection, topology, protocol, resource
10	loop, communication, computation, execution, java, strategy, work, agent, different, programming
11	path, flow, condition, machine, information, face, complexity, dependence, protocol, value
12	cache, database, tree, study, message, process, single, machine, scheme, multiple
13	java, programming, state, scheme, computer, class, core, event, task, proposed
14	flow, region, incremental, software, prefetching, structure, control, optimization, object, case
15	service, business, information, framework, fault, communication, solution, heap, university, flow
16	java, code, compiler, machine, tool, static, software, test, class, virtual
17	programming, structure, optimization, dynamic, environment, tree, visualization, compression, mining, function
18	property, register, software, file, hierarchy, computer, cost, scheduling, class, resource
19	object, collection, user, garbage, collector, type, space, surface, query, code
20	pointer, fortran, information, graph, technology, change, code, retrieval, software, call

Table 2: LDA Topics and Keywords in Cluster 30

of compiler optimization, system architecture, and efficient data processing, showcases a deep engagement with the foundational elements of computing. This reflects a shared interest in enhancing the performance and scalability of computing systems, which is pivotal in advancing computing technologies.

The coherence score of approximately 0.328 for the LDA topics suggests a moderate level of thematic consistency, indicative of well-established research domains within the cluster. By focusing on these influential authors and their related topics, we can gain valuable insights into the core areas of research that have shown development in computing and AI. After identifying influential authors, it is also interesting to look into their collaboration networks, for instance William J. Dally's network shown in Figure 4.

8.3 Large Community

I identified and analyzed five large clusters from the collaborator network, each with over 15,000 papers, while most clusters contained fewer than 15,000 papers. The topic modeling extracted a diverse range of topics from each large cluster. The complete topic modeling results are documented in Large_community_detection.txt. Each large cluster has 180-400 topics extracted based on its size.

Cluster 4 is among the 5 large clusters. In Cluster 4, the topics collectively cover a wide range of areas in computer science, from AI and software development to hardware optimization and image processing, reflecting the diverse interests and research areas within the field. For example, as

shown in table 3, there are some interesting topics within this cluster and potential interpretation of each topic.

8.4 Research Group Finding

In the demonstrated example, the search focused on topics related to 'image', 'video', 'pixels', 'resolution', and 'media'. The search result is in 4

The search yielded several clusters with topics highly relevant to the input keywords. Each cluster comprises topics with a distribution of keywords indicating their focus areas. This process aids users, especially those new to a field, in pinpointing research groups that align with their interests.

This methodology simplifies the process of identifying key research areas and influential authors. Subsequent steps involve pinpointing specific papers and authors within these topics, offering a direct pathway for newcomers to reach out to leading experts in their field of interest.

8.5 Influential Researchers

We analyzed the 5 largest cluster to find influential researchers. The results is in Table 5. If you do some research on these authors, they are mostly renowned professors or researchers. We also visualized subgraph of each cluster with influential researchers highlighted, which is shown in Figure 5, Figure 6, Figure 7, Figure 8, Figure 9.

9 Conclusion

In this project, we analyzed collaborator network constructed from a dataset containing information of ACM journal as of 2010. We identified large research communities, popular research topics, and influential researchers in each community. Furthermore, we provide a simple version of research group finding system, to guide new researchers to find research groups suitable based on their research interest.

10 Future Work

10.1 Temporal Analysis of ACM Publications

Expanding the dataset to include ACM publications from different time periods would allow for a temporal analysis of the academic landscape. This expansion would enable us to:

- **Track Evolution of Research Topics**: By comparing topics from different eras, we can trace the evolution of research interests and technological advancements within computing.
- **Observe Changes in Collaboration Patterns**: Analyzing how author collaboration networks evolve over time could reveal shifts in research methodologies and practices.
- **Identify Emerging Researchers**: Comparing data from different periods can spotlight newly active researchers who are becoming influential in their fields.

Such a temporal analysis would provide a dynamic view of the research field, highlighting trends, transitions, and the emergence of new technologies and theories.

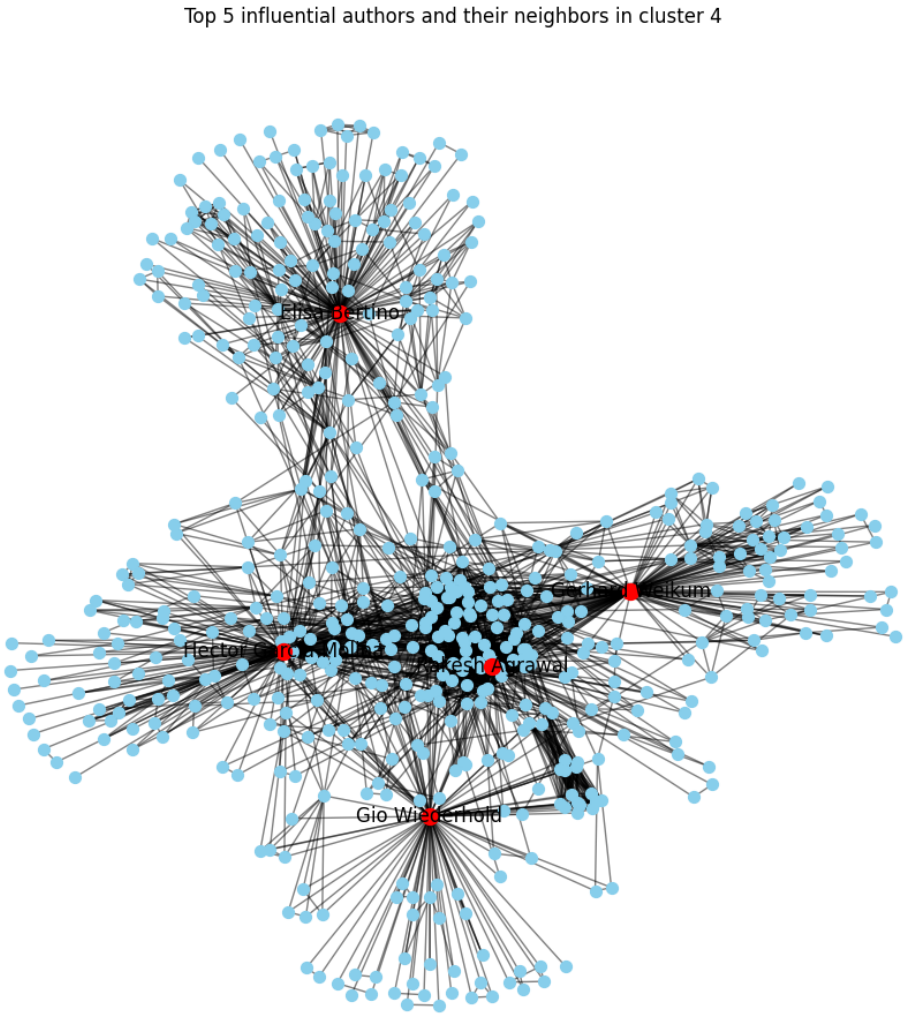

Figure 5: Cluster 4 visualization with top 5 influential authors highlighted in red.

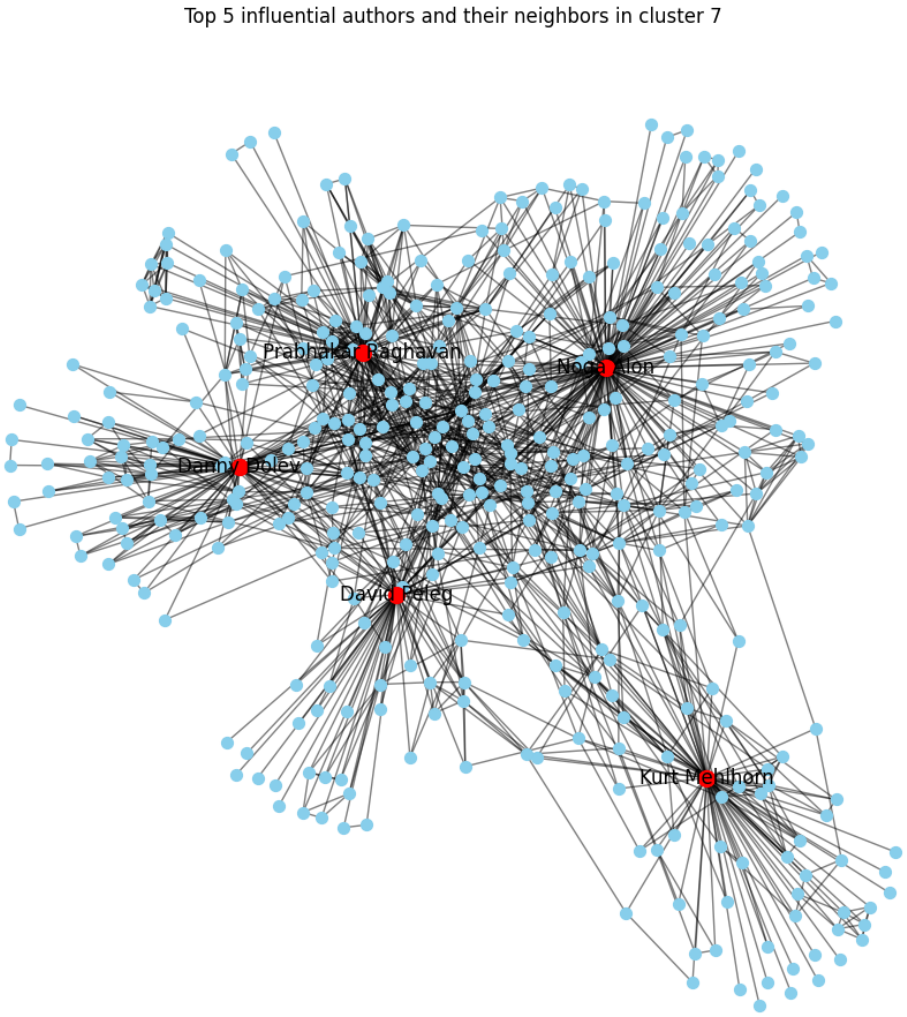

Figure 6: Cluster 7 visualization with top 5 influential authors highlighted in red.

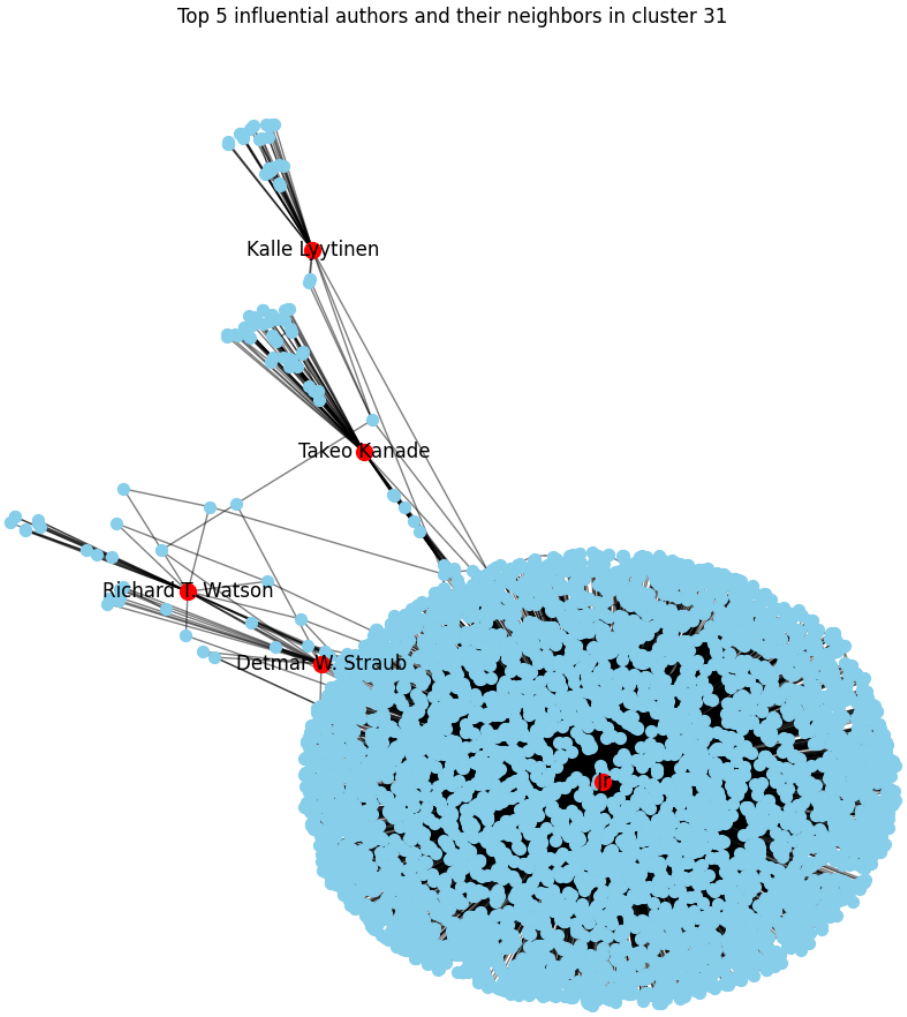

Figure 7: Cluster 31 visualization with top 5 influential authors highlighted in red.

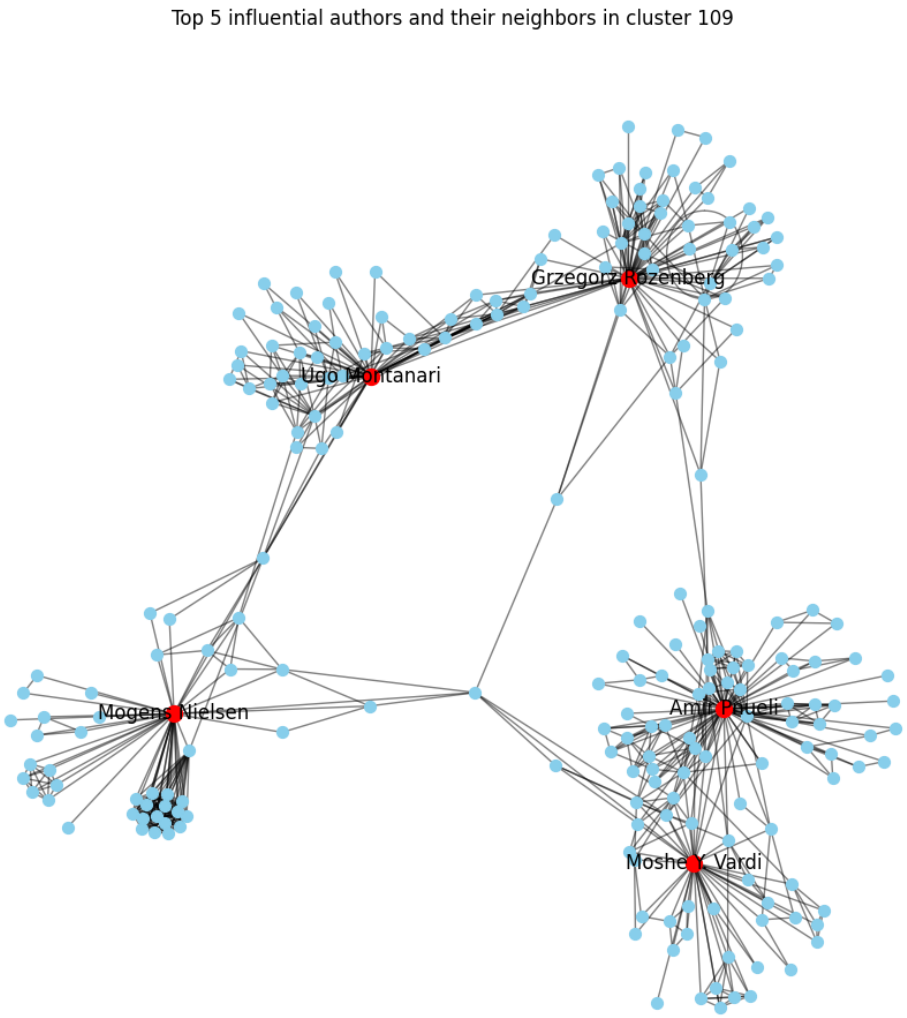

Figure 8: Cluster 109 visualization with top 5 influential authors highlighted in red.

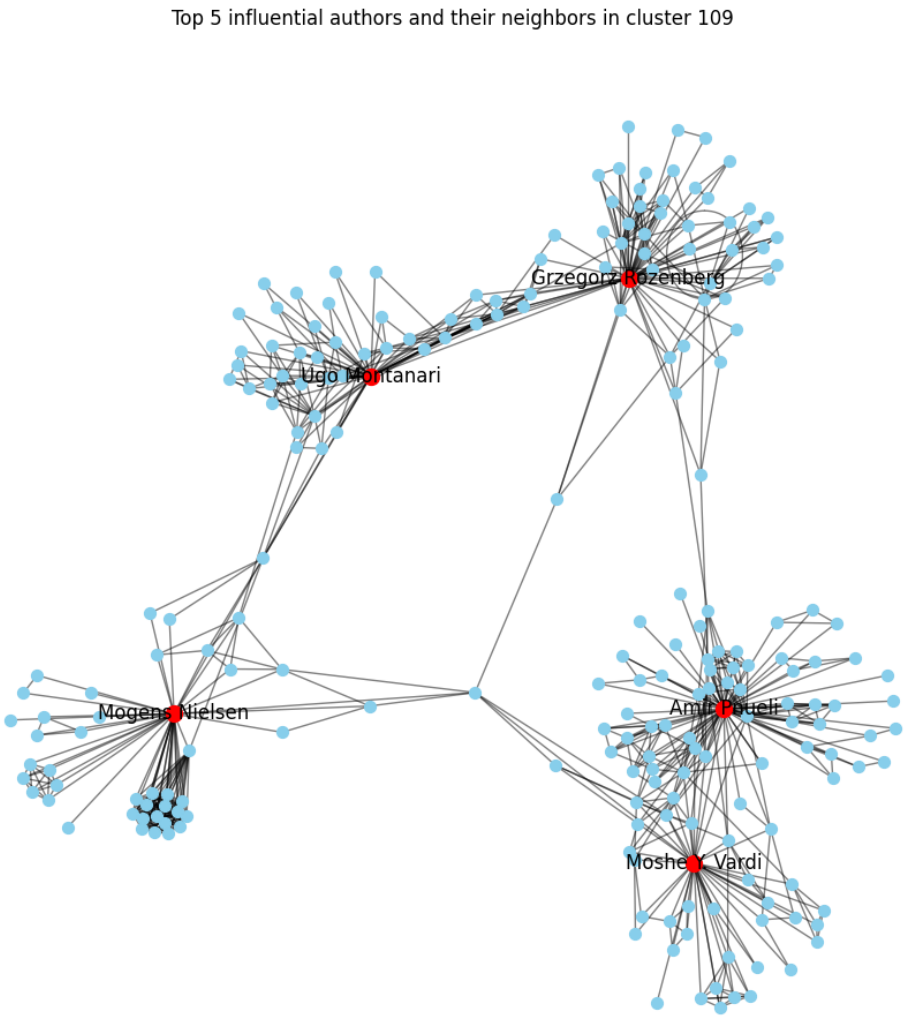

Figure 9: Cluster 149 visualization with top 5 influential authors highlighted in red.

10.2 Incorporating Multiple Journal Datasets

Another significant extension of this work would involve integrating datasets from various academic journals. This broader approach would yield:

- **A More Comprehensive Network Analysis**: By transcending the boundaries of computing machinery, a multi-disciplinary analysis could uncover cross-field collaborations and interdisciplinary research trends.
- **Diverse Research Insights**: Combining datasets from different fields can provide a more holistic view of scientific research, revealing connections and influences across disciplines.

10.3 Further Methodological Enhancements

In addition to expanding the dataset, future work could also focus on methodological improvements and new analytical techniques:

- **Advanced NLP Techniques**: Employing more sophisticated natural language processing methods, such as neural network-based models, could refine topic extraction and sentiment analysis.
- **Citation Network Analysis**: An in-depth examination of citation networks with missing data filled in could yield insights into the impact and influence of specific papers and authors.
- **Predictive Analytics**: Utilizing machine learning algorithms to predict future research trends and emerging areas of study based on historical data.

These methodological enhancements would not only deepen the analysis but also broaden its scope, potentially leading to groundbreaking insights into the nature and trajectory of academic research.

In conclusion, the proposed future work aims to build upon the foundation laid by this project, leveraging larger, more diverse datasets and advanced analytical techniques to gain a richer, more nuanced understanding of the academic landscape across various disciplines and eras.

Topic Number	Keywords	Interpretation
20	Workshop, conference, international, intelligence, proceeding, artificial, 2006, life, working, 2007, growth, 2004, evolving, 2008, academic, held, technology, committee, mine	Focuses on international conferences and workshops in artificial intelligence, highlighting the progression and key developments in AI over various years.
3	Parallel, shared, concurrent, implementation, production, sorting, machine, balanced, priority, execution, switch, percent, usability, passing, similarly, floating, accessibility, serve, sgml, calibration	Related to parallel and concurrent computing, emphasizing shared resources, algorithm implementation, and system optimization for efficiency.
0	Image, content, retrieval, color, proposed, indexing, formulated, similar, latest, enormous, searching, similarity, novel, wavelet, chase, experimental, ultimate, dataflow, presented, essential	Centers on image processing and retrieval, focusing on content management, color indexing, and search algorithms for image databases.
78	Java, coefficient, price, networking, counting, combinatorial, triangle, symbol, auction, converge, mail, besides, black, pseudo, maximizing, room, producer, additionally, chat, hold	Discusses Java programming, with a focus on networking and algorithmic concepts such as counting, pricing, and combinatorial analysis.
127	Environment, architecture, platform, integrated, support, implementation, describe, discus, interoperability, implemented, provides, intended, describes, briefly, different, permit, goal, motivation, exploratory, orientation	Deals with software architecture and environmental platforms, emphasizing integrated systems, platform support, and interoperability in software development.
171	Memory, base, hardware, latency, buffer, main, cost, scan, access, miss, size, tailored, amount, speedup, order, high, store, total, overhead, single	Focuses on computer hardware, particularly memory systems and optimization techniques, including aspects of latency, buffering, and hardware cost.

Table 3: Examples of Detected Topics in Cluster 4

Cluster ID	Topic ID	Top Keywords
4	0	image, content, retrieval, color, proposed
	105	source, integration, integrating, uncertainty
	169	video, tracking, predicate, head, scene
7	174	recognition, series, resolution, closure
	270	image, recursive, graphic, starting
	242	quality, video, unique, maintained
31	132	dynamic, state, resolution, availability
	61	image, value, graphic, productivity
	106	video, post, faculty, date
109	26	resolution, parser, write, video
	34	image, representation, towards
	162	invariant, survey, parallelism
149	222	analysis, image, video, conceptual
	192	theory, decision, procedure
	239	query, database, multimedia

Table 4: Clusters and their corresponding topics with top keywords

Cluster ID	Top 5 Influential Authors
149	Wolfgang Nejdl, Gustaf Neumann, Andrew Taylor, Alexandros Nanopoulos
109	Grzegorz Rozenberg, Mogens Nielsen, Amir Pnueli, Ugo Montanari, Moshe Y. Vardi
31	Jr., Kalle Lyytinen, Detmar W. Straub, Takeo Kanade, Richard T. Watson
7	Noga Alon, Kurt Mehlhorn, Prabhakar Raghavan, Danny Dolev, David Peleg
4	Elisa Bertino, Hector Garcia-Molina, Rakesh Agrawal, Gerhard Weikum, Gio Wiederhold

Table 5: Top Influential Authors in the Largest Clusters