Link Analysis and Prediction; Community Detection

Yash Dani*†
York University
Toronto, Canada
yashdani@my.yorku.ca

Abhinav Syal‡†
York University
Toronto, Canada
syal28@my.yorku.ca

ABSTRACT

This report examines the temporal dynamics and structural properties of co-authorship networks using the DBLP dataset for 2005 and 2006. We generate temporal snapshots, perform link analysis, predict future collaborations, and detect community structures. Our analysis includes PageRank and edge betweenness scores, various link prediction methods, and the Louvain method for community detection. The findings highlight significant patterns and potential predictive models for future collaborations.

KEYWORDS

Link Analysis, Link Prediction, Community Detection, Network Analysis, DBLP Dataset, PageRank, Edge Betweenness, Temporal Snapshots

1 INTRODUCTION

1.1 Objective and Motivation

The objective is to analyze the temporal dynamics and structural properties of the DBLP co-authorship network. This involves generating temporal snapshots, performing link analysis, predicting future collaborations, and detecting community structures to understand evolving co-authorship patterns over time.

1.2 Importance

Analyzing co-authorship networks reveals collaboration patterns, influential researchers, and trends in research. Studying these networks helps understand knowledge dissemination, predict future collaborations, and identify research communities. This analysis is valuable for researchers, institutions, and funding agencies.

1.3 Methods and Datasets

We use the DBLP dataset with (author1, author2, year) triples for co-authorships. Our analysis includes creating temporal graphs for 2005 and 2006, performing link analysis with PageRank and edge betweenness scores, predicting future collaborations using various methods, and detecting communities with the Louvain method. We utilize NetworkX for graph analysis.

2 TEMPORAL GRAPHS

2.1 Preparing the Required Graphs

2.1.1 Common Methodology for Creating the Undirected Unweighted Graphs. To construct the undirected, unweighted graphs for 2005 and 2006, we utilized the DBLP co-authorship dataset, consisting of triples representing co-authorships between two authors in a given year.

- **Loading Data**: Loaded the dataset from a JSON file using the `json` library.
- **Filtering by Year**: Iterated through the dataset, filtering records for 2005 and 2006.
- **Adding Nodes and Edges**: Used NetworkX's `add_node` and `add_edge` methods to create graphs and populate them with filtered records.

2.1.2 Creating Weighted Graphs. To create the weighted graph for 2005, we:

Initialized a dictionary for edge weights. Iterated through the 2005 collaboration network using NetworkX's `add_edge` method to add edges. For each co-authorship, if the edge existed, we incremented its weight by one; otherwise, we initialized the weight to one. After computing all edge weights, we added these weighted edges to the graph.

2.1.3 Extracting Greatest Connected Component (GCC). To focus on the most relevant network portion, we extracted the greatest connected component (GCC) from each graph:

Used NetworkX's `connected_components` to identify all connected components. Selected the largest component using `max` with `key=len`. Created subgraphs of these components with `subgraph` and copied them to preserve the original structure.

Saved these GCCs to .gexf files for further analysis using NetworkX's `write_gexf` method.

2.2 Number of Nodes and Edges

The number of nodes (`.number_of_nodes`) and edges (`.number_of_edges`) in the GCC for each graph is as follows:

- **dblp2005**: 106,943 nodes, 300,043 edges.
- **dblp2006**: 50,879 nodes, 121,822 edges.
- **dblp2005w**: 106,943 nodes, 300,043 edges.

```
The number of nodes in the GCC of 2005 is: 106943
The number of edges in the GCC of 2005 is: 300043
The number of nodes in the GCC of 2005_weighted is: 106943
The number of edges in the GCC of 2005_weighted is: 300043
The number of nodes in the GCC of 2006 is: 50879
The number of edges in the GCC of 2006 is: 121822
```

Figure 1: Console Output for Number of Nodes and Edges in GCC of the required graphs

3 NODE AND EDGE IMPORTANCE IN GRAPHS

3.1 PageRank Scores

PageRank is an algorithm originally used by Google to rank web pages in their search engine results. It measures the importance of nodes (authors) in a graph based on incoming links. It has two main interpretations:

*Yash's Student ID: 218465231
†Both authors contributed equally to this assignment.
‡Abhinav's Student ID: 218820330

Flow Interpretation: PageRank is distributed to neighbors based on their importance. Nodes with more and higher-ranked incoming connections have higher ranks.

Random Walk Interpretation: A random walker moves through the nodes following edges. The PageRank score represents the probability of the walker being at a node, considering direct and indirect links.

3.1.1 Calculation Method. We calculated the PageRank scores for the graphs dblp2005, dblp2006, and dblp2005w using the `pagerank` function from NetworkX with a damping factor α of 0.85 and up to 100 iterations. For the weighted graph, edge weights were considered.

The scores were rounded to three significant figures, and the top 50 authors were selected for each graph. These scores were saved in text files (e.g., 2005_pagerank_scores.txt).

```
PageRank scores written to 2005_pagerank_scores.txt
Edge betweenness scores and authors written to 2005_edge_betweenness_scores.txt
```

Figure 2: Console Output confirming file saved

3.1.2 Results. The top 50 authors based on PageRank scores were identified and stored for each graph.

```
dblp2005:
Alex Panaretos: 1.419e-04
Alessio Mazzanti: 1.331e-04
Andrew Fish: 1.184e-04
Daniel Berg: 1.123e-04
Atílio E. Reggiani: 1.05e-04
Haijin Yan: 1.024e-04
ByungMoon Kim: 9.97e-05
Amitava Sen: 9.74e-05
Adrian Price: 9.73e-05
Dayong Li: 9.71e-05
Chon-in Wu: 9.63e-05
Adriane Swalm Durey: 9.51e-05
Duc Minh Nguyen: 9.5e-05
Al Wagner: 9.29e-05
A. Gonzalez: 9.16e-05
Andreas Kassler: 9.09e-05
Arnaud Membre: 9.06e-05
Andrew S. Karson: 9.05e-05
Ashikur Rahman: 8.96e-05
Amol Pednekar: 8.92e-05
Abhijeet Joglekar: 8.77e-05
Albert A. Angehrn: 8.76e-05
Christian Fritz 0001: 8.76e-05
Hongxi Wang: 8.65e-05
Amir Sheikh Zeineddini: 8.63e-05
Avi Wigderson: 8.52e-05
Avner Bar-Hen: 8.39e-05
Andrea Tagarelli: 8.36e-05
Alan Martin: 8.35e-05
Arnoud C. Klaren: 8.28e-05
B. Venkata Ramana: 8.26e-05
Corneliu Henegar: 8.15e-05
Ayman F. Habib: 8.1e-05
A. Masoudnia: 8.09e-05
Adam T. Sampson: 8.03e-05
Alexandra Volanschi: 8.e-05
Chang-Jung Ku: 7.98e-05
B. Roe Hemenway: 7.95e-05
Ben Kobler: 7.9e-05
Chi-Leung Wong: 7.86e-05
A. J. P. Brown: 7.85e-05
Carlos Bernal Ruiz: 7.85e-05
Eduardo S. Vera: 7.84e-05
A. Katz: 7.83e-05
Alex K. Simpson: 7.8e-05
Alessandro Giua: 7.79e-05
Alev Kaya: 7.72e-05
Günther Nürnberger: 7.71e-05
A. Keller: 7.67e-05
Andrea Del Re: 7.63e-05

dblp2006:
Antonio J. Dorta: 2.372e-04
Xinye Cai: 2.214e-04
Yasuaki Kakehi: 1.93e-04
Martin Marinov: 1.862e-04
Wolfgang Helmberg: 1.774e-04
Shaya Potter: 1.77e-04
Kohei Ohtake: 1.712e-04
Heng Huang: 1.685e-04
Gemma L. Holliday: 1.677e-04
Sira E. Palazuelos-Cagigas: 1.654e-04
Philippas Tsigas: 1.645e-04
Elke Achtert: 1.597e-04
Haixun Wang: 1.589e-04
Alin Dobra: 1.561e-04
Joseph Pizzimenti: 1.55e-04
Zhen Li: 1.54e-04
Olivier Chevassut: 1.528e-04
Jose Neves: 1.517e-04
Quan Huynh-Thu: 1.503e-04
Steven K. Reinhardt: 1.487e-04
John Sharry: 1.476e-04
Raghavendra Chandrashekara: 1.465e-04
Kimmo Fredriksson: 1.434e-04
Lorenzo Marconi: 1.423e-04
Wenxin Liu: 1.423e-04
Tatsuro Takahashi: 1.398e-04
George M. Tzoumas: 1.397e-04
Shin-ichi Tanigawa: 1.396e-04
Anurag Kumar 0001: 1.387e-04
Seungji Yang: 1.376e-04
Jan Cornelis: 1.368e-04
Peter C. Nelson: 1.363e-04
Ana L. N. Fred: 1.361e-04
Luca Cardelli: 1.326e-04
Karlis Kaugars: 1.312e-04
Shibu Menon: 1.312e-04
Amitabh Varshney: 1.299e-04
Junjuan Xu: 1.294e-04
Mike Carbonaro: 1.27e-04
Li Zou: 1.267e-04
Sana Sfar: 1.266e-04
Lining Sun: 1.265e-04
Luca Trevisan: 1.257e-04
Ishai Rabinovitz: 1.255e-04
Wei-Yun Yau: 1.246e-04
Aske Simon Christensen: 1.245e-04
Huiping Cao: 1.235e-04
Pierre Hellier: 1.22e-04
Richard Mayr: 1.217e-04
Joshua D. Knowles: 1.213e-04

dblp2005w:
Alex Panaretos: 1.419e-04
Alessio Mazzanti: 1.331e-04
Andrew Fish: 1.184e-04
Daniel Berg: 1.123e-04
Atelio E. Reggiani: 1.05e-04
Haijin Yan: 1.024e-04
ByungMoon Kim: 9.97e-05
Amitava Sen: 9.74e-05
Adrian Price: 9.73e-05
Dayong Li: 9.71e-05
Chon-in Wu: 9.63e-05
Adriane Swalm Durey: 9.51e-05
Duc Minh Nguyen: 9.5e-05
Al Wagner: 9.29e-05
A. Gonzalez: 9.16e-05
Andreas Kassler: 9.09e-05
Arnaud Membre: 9.06e-05
Andrew S. Karson: 9.05e-05
Ashikur Rahman: 8.96e-05
Amol Pednekar: 8.92e-05
Abhijeet Joglekar: 8.77e-05
Albert A. Angehrn: 8.76e-05
Christian Fritz 0001: 8.76e-05
Hongxi Wang: 8.65e-05
Amir Sheikh Zeineddini: 8.63e-05
Avi Wigderson: 8.52e-05
Avner Bar-Hen: 8.39e-05
Andrea Tagarelli: 8.36e-05
Alan Martin: 8.35e-05
Arnoud C. Klaren: 8.28e-05
B. Venkata Ramana: 8.26e-05
Corneliu Henegar: 8.15e-05
Ayman F. Habib: 8.1e-05
A. Masoudnia: 8.09e-05
Adam T. Sampson: 8.03e-05
Alexandra Volanschi: 8.e-05
Chang-Jung Ku: 7.98e-05
B. Roe Hemenway: 7.95e-05
Ben Kobler: 7.9e-05
Chi-Leung Wong: 7.86e-05
A. J. P. Brown: 7.85e-05
Carlos Bernal Ruiz: 7.85e-05
Eduardo S. Vera: 7.84e-05
A. Katz: 7.83e-05
Alex K. Simpson: 7.8e-05
Alessandro Giua: 7.79e-05
Alev Kaya: 7.72e-05
Günther Nürnberger: 7.71e-05
A. Keller: 7.67e-05
Andrea Del Re: 7.63e-05
```

3.1.3 Commentary. The PageRank scores for the dblp2005, dblp2006, and dblp2005w graphs provide valuable insights into the co-authorship networks.

dblp2005 vs. dblp2006: Top authors in dblp2006 have higher PageRank scores than in dblp2005, suggesting a more interconnected network. For example, Antonio J. Dorta in dblp2006 has 2.372×10^{-4}, while Alex Panaretos in dblp2005 has 1.419×10^{-4}, indicating a higher concentration of influential authors in 2006.

dblp2005 vs. dblp2005w: Similar rankings in both unweighted and weighted 2005 graphs imply that the number of co-authored papers (weights) does not significantly alter overall rankings. Alex Panaretos remains the top author in both graphs, showing consistent influence irrespective of edge weights.

Distribution of Scores: The wider range and higher peaks in PageRank scores for dblp2006 reflect a more pronounced hierarchical structure and centralization, likely due to evolving co-authorship patterns.

Significance of Top Authors: Higher scores of top authors in both years highlight their strong presence and influence in the academic community, driven by extensive collaborations and multiple publications.

Overall, 2006's network is more concentrated with influential authors, and edge weights in 2005 offer a nuanced but non-transformative view of author influence.

3.2 Edge Betweenness Scores

Edge betweenness centrality measures the importance of edges in a graph by quantifying the number of shortest paths that pass through an edge. Higher edge betweenness indicates an edge's crucial role in maintaining network connectivity.

3.2.1 Calculation Method. To calculate edge betweenness scores for dblp2005, dblp2006, and dblp2005w, we used the `edge_betweenness_centrality` function from the NetworkX library. This computes shortest paths and determines edge centrality based on their frequency in these paths. Scores were rounded to four significant figures and sorted to identify the top 20 edges with the highest centrality.

3.2.2 Results. The 20 most important edges (`pairs of author names`) based on edge betweenness scores were identified and stored for each graph.

```
dblp2005:
Li Jin – Hsien-Huang P. Wu: 9.36e-06
Ming-Hwa Sheu – Hsien-Huang P. Wu: 9.36e-06
Chao Zhang – Zhenghui Gui: 9.35e-06
Yong Fang – Wei Fang: 9.35e-06
Alex Pang – David Kao: 9.35e-06
David Kao – David H. Liang: 9.35e-06
Bastian Wormuth – Peter W. Eklund: 9.35e-06
Ming-Hwa Sheu – Chishyan Liaw: 9.35e-06
Chishyan Liaw – Cherng-yue Huang: 9.35e-06
Peter W. Eklund – Richard Cole: 9.13e-06
Rajnikant V. Patel – Heidar A. Talebi: 8.76e-06
Marco Cristo – Paulo Braz Golgher: 8.21e-06
Rajnikant V. Patel – Jorge Angeles: 8.18e-06
Heng Wang – Jorge Angeles: 7.41e-06
Tao Wang – Li Jin: 6.99e-06
Andrew Y. Ng – Jeff Michels: 6.34e-06
Dennis Shasha – Richard Cole: 5.3e-06
Marcos André Gonçalves – Karla A. V. Borges: 5.21e-06
Alex Pang – Alisa Neeman: 4.99e-06
Weiguo Fan – Marco Cristo: 4.97e-06

dblp2006:
Tamio Arai – Kazunori Umeda: 1.83e-05
Marc Rioux – Kazunori Umeda: 1.829e-05
Marc Rioux – Richard Lepage: 1.829e-05
Sang-Cheol Park – Soo-Hyung Kim: 1.829e-05
Seong-Whan Lee – Mohamed Kamel: 1.791e-05
Seong-Whan Lee – Sang-Cheol Park: 1.791e-05
Marc Shapiro – Luis Rodrigues: 1.482e-05
Hua Li – Jie Chen 0002: 1.368e-05
Al Brown – Alun D. Preece: 1.333e-05
Yuanchun Shi – Yu Chen: 1.085e-05
Nicholas R. Jennings – Alun D. Preece: 9.39e-06
Chun Tung Chou – Sanjay Jha: 8.95e-06
Yan Yang – Mohamed Kamel: 8.83e-06
Tamio Arai – Takashi Sato: 7.03e-06
Masanori Hashimoto – Takashi Sato: 6.99e-06
Masanori Hashimoto – Yun Yang: 6.91e-06
Chun Tung Chou – Wei Liu: 6.87e-06
David Zhang – Mohamed Kamel: 6.8e-06
Gerhard Weikum – Luis Rodrigues: 6.64e-06
Michael L. Scott – Galen C. Hunt: 6.3e-06

dblp2005w:
Christoph Schlieder – Klaus Stein: 9.35e-06
Selim G. Akl – Marius Nagy: 9.35e-06
Hermann de Meer – Christian Koppen: 9.35e-06
Klaus Stein – Claudia Hess: 9.35e-06
Koji Kajiwara – Hirokazu Yamamoto: 9.29e-06
Hirokazu Yamamoto – Tomoaki Miura: 9.26e-06
Tomoaki Miura – Alfredo R. Huete: 9.26e-06
David Nister – Steven C. Hsu: 9.17e-06
Rakesh Kumar 0001 – Steven C. Hsu: 9.16e-06
Alfredo R. Huete – Bin Tan: 8.32e-06
Sanjeeb Bhoi – Yong Xie: 7.85e-06
David Nister – Frederik Schaffalitzky: 7.63e-06
Thomas C. Rindflesch – Vincent R. Sanchez: 6.84e-06
Wei-Ying Ma – Hui-Min Yan: 6.82e-06
Ellen W. Zegura – Ruomei Gao: 5.98e-06
Christoph Schlieder – Markus Knauff: 5.75e-06
Hsinchun Chen – Gondy Leroy: 5.4e-06
Gondy Leroy – Thomas C. Rindflesch: 5.4e-06
Richard Wright – Franciska de Jong: 3.91e-06
David Hutchison – Hermann de Meer: 3.66e-06
```

3.2.3 Commentary. The edge betweenness scores for dblp2005, dblp2006, and dblp2005w reveal critical connections within these co-authorship networks.

dblp2005 vs. dblp2006: The top edges in dblp2006 have higher betweenness scores, indicating more prominent bridges in the 2006 network. The highest score in dblp2006 is 1.83×10^{-5} compared to 9.36×10^{-6} in dblp2005, suggesting a more collaborative network in 2006.

dblp2005 vs. dblp2005w: The weighted graph (dblp2005w) shows similar top edges to the unweighted graph (dblp2005), with slight differences in scores. The highest edge betweenness in both graphs is 9.35×10^{-6}, indicating weights do not significantly alter key bridging edges.

Distribution of Scores: dblp2006 shows a broader range of edge betweenness scores, reflecting a more hierarchical network compared to dblp2005. This could be due to an increase in collaborations or more complex interactions in 2006.

Significance of Top Edges: The top edges in all graphs are critical for facilitating information flow and collaboration. Consistency of certain edges across unweighted and weighted graphs highlights their importance, irrespective of the number of co-authored papers.

Overall, dblp2006 shows a more interconnected structure, and the weighted edges in dblp2005w provide a nuanced view of influential connections.

4 LINK PREDICTION IN GRAPHS

Link prediction aims to predict future interactions (edges) in a network based on its current structure. We use the co-authorship network from 2005 to predict new co-authorships that will appear in 2006.

4.1 Core Graph Construction

4.1.1 Method. We created the dblp2005-core and dblp2006-core graphs by filtering nodes from the 2005 and 2006 co-authorship networks, respectively, to include only those with a degree ≥ 3.

- Calculated node degrees using the `degree` method in the 2005 and 2006 GCCs.

- Created new graphs (`dblp2005_core` and `dblp2006_core`) including nodes with degree ≥ 3 by iterating over node degrees and adding edges with `add_node` and `add_edge` methods from NetworkX.

4.1.2 Results.

- dblp2005-core: 77,153 nodes, 255,815 edges.
- dblp2006-core: 32,948 nodes, 97,478 edges.

```
The number of nodes in the dblp2005_core is: 77153
The number of edges in the dblp2005_core is: 255815
The number of nodes in the dblp2006_core is: 32948
The number of edges in the dblp2006_core is: 97478
The number of target edges is:  29578
```

Figure 3: Core graph construction results for dblp2005 and dblp2006.

4.2 FoF Computation

Computed the list of friends-of-friends (FoF) in dblp2005-core.

To identify potential new collaborations, we computed the list of friends-of-friends (FoF) within the dblp2005-core graph. The process involved:

- **Initialization**: Initialized an empty set for FoF pairs.
- **Iterating over Nodes**: For each node in dblp2005-core, identified its neighbors.
- **Identifying Common Neighbors**: For each neighbor's neighbors, checked for common neighbors. If found and not the original node, added the pair to the FoF set.

```
# Initialize the set of friends-of-friends (FoF)
fof = set()

# Compute the set of friends-of-friends (FoF)
for node1 in dblp2005_core.nodes():
    neighbors_node1 = [node for node in dblp2005_core.neighbors(node1)]
    for neighbor in neighbors_node1:
        neighbors_neighbor = [node for node in dblp2005_core.neighbors(neighbor)
            ]
        for common_neighbor in set(neighbors_node1) & set(neighbors_neighbor):
            if common_neighbor != node1 and (node1, common_neighbor) not in fof:
                fof.add((node1, common_neighbor))
```

This method ensures that all pairs of nodes sharing a common neighbor (2 hops away) are identified, providing a basis for predicting potential new collaborations. The set is converted to a list, filtered for duplicates, and saved to a .txt file for future use.

4.3 Target Edges Derivation

To identify the new co-authorships formed in 2006, we compute the set of edges that do not exist in dblp2005-core but exist in dblp2006-core, using set operations to find the difference between the edge sets. Please refer to Fig. 3 for console output.

4.4 Prediction Methods and Computation

Link prediction methods estimate the likelihood of future links between nodes in a network. We implemented and evaluated the following methods:

- **Random Predictor (RD)**: Randomly predicts links between nodes.
- **Common Neighbors (CN)**: Scores based on the number of shared neighbors between nodes.
- **Jaccard Coefficient (JC)**: Calculates the probability of shared features, defined as the intersection size divided by the union size of neighbors.
- **Preferential Attachment (PA)**: Predicts new links are more likely with nodes having a high degree.
- **Adamic/Adar (AA)**: Weighs common neighbors with fewer neighbors more heavily, giving rare features more weight.

4.4.1 Computation Methodology. We computed the set of predicted edges P using a custom loop implemented to efficiently handle the Friends-of-Friends (FoF) computations specific to our application, for each prediction method. This approach proved faster and more effective than existing methods.

```
# Change the graph into a DiGraph to find the out_degrees
G = nx.DiGraph(dblp2005_core)

# Out degrees of each node
out_degrees = dict(G.out_degree)

# Find the maximum out-degree
max_out_degree = max(out_degrees.values())

# Dictionaries of scores
fof_common_neighbor_scores = {}
jaccard_scores = {}
pas_scores = {}
adamic_scores = {}

# Compute scores for each method
for node1, node2 in fof:
    # Common Neighbor Scores
    common_neighbor_intersection = set(dblp2005_core.neighbors(node1)) &
                                   set(dblp2005_core.neighbors(node2))
    fof_common_neighbor_score = len(common_neighbor_intersection)
    fof_common_neighbor_scores[(node1, node2)] = fof_common_neighbor_score

    # Jaccard Neighbor Scores
    common_neighbor_union = set(dblp2005_core.neighbors(node1)) |
                            set(dblp2005_core.neighbors(node2))
    fof_common_neighbor_union = len(common_neighbor_union)
    jaccard_coefficient = fof_common_neighbor_score / fof_common_neighbor_union
    jaccard_scores[(node1, node2)] = jaccard_coefficient

    # Preferential Attachment Scores
    pas_score = (math.log(out_degrees[node1]) + math.log(out_degrees[node2])) /
                math.log(max_out_degree)
    pas_scores[(node1, node2)] = pas_score

    # Adamic-Adar Scores
    adamic_score = 0
    for neighbor in common_neighbor_intersection:
        if out_degrees[neighbor] == 0:
            adamic_score += 0
        else:
            adamic_score += 1 / math.log(out_degrees[neighbor])
    adamic_scores[(node1, node2)] = adamic_score
```

In summary, we used common neighbors, Jaccard coefficient, preferential attachment, and Adamic/Adar methods to compute scores for potential edges in the dblp2005-core graph, which were then used to predict the likelihood of new co-authorships in 2006.

4.5 Precision at k

The precision at k evaluates the effectiveness of link prediction methods. For each k value (10, 20, 50, 100, and the total number of target edges), a random subset of Friends-of-Friends (FoF) is selected to predict new edges. Precision is derived by comparing these predicted edges with the actual target edges. The code iterates through each k value and computes precision for different methods: Random Predictor (RD), Common Neighbors (CN), Jaccard Coefficient (JC), Preferential Attachment (PA), and Adamic/Adar (AA). For each method, the top k scores are identified, converted into a set of predicted edges, and precision is calculated as the ratio of correctly predicted edges to the total predicted edges.

4.6 Results and Analysis

4.6.1 Results.

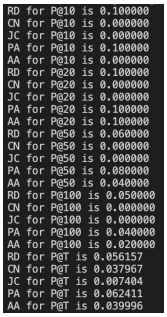

Figure 4: Precision scores for each prediction method

4.6.2 Analysis. The precision scores for the different prediction methods reveal key insights:

- **Random Predictor (RD)**: Generally low precision, decreasing with higher k values, indicating random guessing is ineffective for predicting new co-authorships.
- **Common Neighbors (CN)**: Precision score of 0 across all k values, suggesting this method is ineffective in the dblp2005-core context.
- **Jaccard Coefficient (JC)**: Similar to CN, it shows a precision score of 0 across all k values, indicating poor performance for this dataset.
- **Preferential Attachment (PA)**: Shows positive results for smaller k values but declines as k increases, indicating it is more effective for predicting fewer new edges.
- **Adamic/Adar (AA)**: Performs better than most methods, particularly for smaller k values, maintaining reasonable precision, indicating robustness in predicting new collaborations.

In summary, Adamic/Adar and Preferential Attachment methods are more effective for link prediction in co-authorship networks, especially for smaller sets of predictions. Conversely, Common Neighbors and Jaccard Coefficient methods perform poorly in this dataset.

5 COMMUNITY DETECTION

5.1 Method

We applied the Louvain method for community detection on the dblp2005 graph. The Girvan-Newman method, which removes edges with the highest betweenness to iteratively derive communities, was deemed unsuitable due to the large network size and computational constraints.

- **Louvain Method**: We utilized the `community.louvain_communities` function from NetworkX to detect communities. The Louvain method is more scalable for large networks compared to Girvan-Newman.
- **Community Sizes**: After partitioning the network into communities, we computed the sizes of these communities. The sizes were sorted in descending order to identify the largest communities.

The Louvain method was chosen for its efficiency in handling large networks, enabling effective community detection in the dblp2005 graph, which is crucial for our analysis

5.2 Results

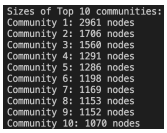

Figure 5: Sizes of the Top 10 Communities in dblp2005

5.3 Commentary

The community sizes in the dblp2005 graph, identified using the Louvain method, reveal a highly modular structure with significant variation, from 2961 nodes in the largest community to 1070 nodes in the smallest of the top 10. This variation indicates diverse collaboration patterns, with tightly-knit groups and broader, less cohesive networks. The largest community likely represents a core research area or closely related topics, while smaller communities are essential for understanding niche areas and emerging fields.

This distribution highlights the interdisciplinary nature of academic research, where substantial collaboration occurs across various domains. Understanding these communities can identify key influencers and potential bridges between research areas, fostering targeted collaboration and resource allocation strategies. The analysis underscores the importance of modular structures in understanding the academic landscape and the evolution of co-authorship networks.

6 CONCLUSION

This report analyzed the DBLP co-authorship network for 2005 and 2006, focusing on temporal dynamics and structural properties. We generated temporal snapshots, performed link analysis, predicted future collaborations, and detected community structures. Key findings include identifying influential authors and crucial connections by calculating PageRank and edge betweenness scores, with the 2006 network showing higher interconnectivity and influence. Evaluating link prediction methods, we found Adamic/Adar and Preferential Attachment to be more effective, while Common Neighbors and Jaccard Coefficient performed poorly.

Using the Louvain method, we identified significant communities in the dblp2005 graph, revealing a varied structure of tightly-knit groups and broader networks. Our analysis provided valuable insights into the evolution and structure of co-authorship networks, aiding researchers, institutions, and funding agencies in fostering productive research collaborations and enhancing academic research growth.

7 REFERENCES

- https://networkx.org/
- http://projects.csail.mit.edu/dnd/DBLP/
- https://perso.uclouvain.be/vincent.blondel/research/louvain.html