UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Graph-Based Entity Resolution and Completion for Academic Knowledge Graphs

Author:
Madison E. CHESTER

Supervisor:
Dr. Dimitri MARINELLI
Dr. Albert DIAZ-GUILERA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2024

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Graph-Based Entity Resolution and Completion for Academic Knowledge Graphs

by Madison E. CHESTER

This thesis explores graph-based entity resolution and completion within academic knowledge graphs, focusing on the complex relationships between authors and papers and between papers themselves using Graph Neural Networks (GNNs). Raw data sourced from the American Physical Society underwent meticulous data cleaning and entity resolution analysis to prepare it for the proposed network. Author grouping strategies and citation overlap were examined, revealing distinct clusters of researchers and insightful patterns in citation relationships. A GNN model was developed using `SAGEConv` layers and heterogeneous transformations to capture local graph structures for accurate link prediction. This model was optimized with mini-batch loading and edge-level splits, which contributed to its high accuracy in predicting links between authors and papers, as demonstrated in the evaluation. The findings underscore the model's capability to uncover hidden relationships and trends within the academic graph. Future work could enhance the model by incorporating additional features, experimenting with alternative GNN architectures, and including more detailed citation contexts and collaboration networks. Overall, this thesis highlights the transformative potential of GNNs in entity resolution and completion for academic knowledge graphs.

Acknowledgements

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dimitri Marinelli and Albert Diaz-Guilera, for their invaluable guidance and encouragement. Their insightful feedback and continuous mentorship were instrumental to the completion of this thesis.

I am also profoundly grateful to my family for their unwavering support and belief in my ability to pursue any academic endeavor. Their patience, understanding, and love have been a constant source of strength and motivation during my master's studies.

Furthermore, I would like to extend my appreciation to my classmates for their camaraderie and collaborative spirit. The shared experiences and collective efforts have greatly enriched my academic experience. Special thanks to Daphne and Jaume, my closest confidants.

Contents

Chapter 1

Introduction

In recent years, the advent of graph neural networks (GNNs) has revolutionized the analysis and utilization of complex relationships and structures in data. This thesis focuses on the development and evaluation of a GNN model for link prediction within an academic knowledge graph. The aim is to predict the likelihood of relationships, such as authorship and citations, between various entities, specifically authors and papers, within the academic domain.

Knowledge graphs represent data in a highly interconnected manner, capturing the rich relationships between entities. In academic knowledge graphs, these entities can include authors, papers, and the citation relationships between them. Accurate link prediction in such graphs has significant implications for a variety of applications, including recommender systems, information retrieval, and understanding the evolution of academic fields. Traditional methods often fall short in capturing the intricate dependencies and multi-relational nature of these graphs, which underscores the need for advanced techniques like GNNs (Kipf and Welling, 2016).

GNNs have emerged as a powerful tool for learning on graph-structured data, leveraging the connectivity patterns to enhance predictive performance. By using neural networks to capture the dependencies between nodes, GNNs can effectively model the complex relationships inherent in knowledge graphs. This capability makes them particularly suitable for tasks such as graph completion, where the goal is to infer missing links or predict future ones based on observed data. In the context of academic knowledge graphs, this can lead to the discovery of new potential collaborations, the identification of influential works, and the mapping of evolving research trends.

The primary objectives of this thesis are twofold: first, to explore and analyze the structure of an academic knowledge graph derived from real-world data; and second, to develop and evaluate a novel GNN model tailored for link prediction tasks within this graph.

The dataset used from the American Physical Society consists of tabular data representing authors, papers, and their relationships extracted from academic databases. The pre-processing steps to convert this tabular data into a structured knowledge graph are detailed, including data cleaning, entity resolution, and integration of citation relationships. Emphasis is placed on ensuring data integrity and preparing the graph for subsequent analysis and modeling.

Author grouping strategies and citation overlap analysis are then delved into, which are critical for understanding the connectivity patterns within the knowledge

graph. Citation overlap analysis examines the degree of similarity between papers based on their citation profiles.

The final chapter focuses on the development and implementation of the GNN model for link prediction. The model architecture, including the use of `SAGEConv` layers and heterogeneous transformations, is detailed. `SAGEConv` (GraphSAGE Convolutional Layer) aggregates information from a node's local neighborhood, allowing the model to learn representations that capture the local graph structure effectively. The model is trained using efficient data handling techniques such as mini-batch loading and edge-level splits to accommodate large-scale heterogeneous graphs.

The training procedures and evaluation metrics for assessing the model's performance are outlined. These procedures offer a comprehensive evaluation of the model's ability to predict links accurately and reliably, considering the diverse relationships and node types within the academic knowledge graph. Finally, a detailed analysis of the experimental results is presented, highlighting the strengths and limitations of the proposed approach.

The code supporting this thesis can be found in the following GitHub repository: `Graph-Based Entity Resolution and Completion for Academic Knowledge Graphs`.

Chapter 2

Background

2.1 Contents

In this chapter, the concept and construction of knowledge graphs are explored, with a particular focus on academic knowledge graphs. Knowledge graphs are structured representations of information where entities, such as authors and papers, are connected by relationships, such as authorship and citation, in a graph format. These graphs integrate diverse types of data into a unified framework, facilitating data retrieval and knowledge discovery. Academic knowledge graphs capture bibliometric data, including publications, authors, affiliations, and research topics, to support advanced analyses like author disambiguation, citation analysis, and collaboration discovery. The methodology for creating these graphs is discussed, encompassing data collection from multiple sources, data integration, and entity resolution techniques. Additionally, the application of machine learning methods to enhance entity resolution and graph completion is delved into. Techniques such as link prediction, node classification, and graph embeddings are highlighted for their roles in enriching graph utility and uncovering hidden relationships within academic research landscapes.

2.2 Knowledge Graphs

Knowledge graphs are structured representations of information, where entities (nodes) are connected by relationships (edges) in a graph format. They are designed to encode and integrate diverse types of data into a unified framework, facilitating data retrieval, reasoning, and knowledge discovery. Knowledge graphs can represent a vast array of domains, including general knowledge (such as Google's Knowledge Graph), domain-specific knowledge (such as medical knowledge graphs), and organizational knowledge (such as enterprise knowledge graphs) (Hogan et al., 2021).

These graphs leverage ontologies and schemas to define entity and relationship types, enabling semantic queries and inference. For instance, a knowledge graph can represent an academic paper as an entity, connected to author entities, institution entities, and other papers through relationships like "written by," "affiliated with," and "cites" (Ehrlinger and Wöß, 2016). This structured representation allows for the application of graph algorithms to extract meaningful insights, such as identifying influential papers, detecting research trends, and uncovering hidden relationships between different research areas (Ji et al., 2021).

2.3 Academic Knowledge Graphs

Academic knowledge graphs are specialized in the academic domain, capturing information about publications, authors, affiliations, research topics, and their interconnections. These graphs are constructed by integrating bibliometric data from various sources, including digital libraries, journal databases, conference proceedings, institutional repositories, and open academic data sources such as OpenAlex. OpenAlex is a distinctly large and openly accessible dataset that provides structured information on scholarly works, authors, institutions, and more, thereby facilitating the creation and enrichment of academic knowledge graphs (OpenAlex, 2024). Although OpenAlex is a leading example of an academic knowledge graph, our analyses use data from the American Physical Society database due to the high level of noise in OpenAlex data. The value of academic knowledge graphs lies in their ability to link diverse entities and reveal complex relationships, which supports advanced bibliometric analyses and enhances academic search engines (Tang et al., 2008).

The nodes in academic knowledge graphs typically represent entities such as papers, authors, institutions, and research topics, while the edges represent relationships like authorship, citation, and co-authorship. By organizing and connecting these entities, academic knowledge graphs support various analytical tasks, including author disambiguation, citation analysis, research trend identification, and collaboration discovery. For instance, citation networks, a subset of academic knowledge graphs, have been extensively used to study the impact and influence of research works (Radicchi, Fortunato, and Vespignani, 2011).

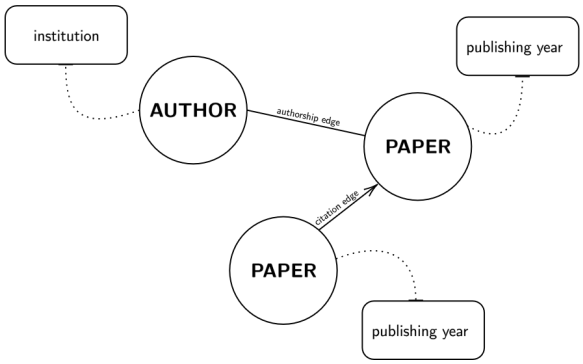

FIGURE 2.1: Visualization of Academic Knowledge Graph. Nodes represent authors and papers, edges represent relationships such as authorship (undirected) and citation (directed). Author features include institution, and paper features include publishing year (Xu et al., 2020).

2.4 Methodology for Creating Academic Knowledge Graphs

Creating an academic knowledge graph involves several key steps:

2.4.1 Data Collection

The first step is aggregating bibliometric data from multiple sources. This involves accessing and extracting metadata from publication databases, digital libraries, and

other repositories. Commonly used data sources include PubMed, IEEE Xplore, arXiv, Google Scholar, and OpenAlex. Each source provides different types of metadata, such as titles, abstracts, authors, affiliations, and citation counts. Collecting data from multiple sources helps in capturing a thorough set of publications and their attributes.

2.4.2 Data Integration

Data integration involves merging data from various sources to create a unified dataset. This process addresses inconsistencies and redundancies, such as different formats for author names or variations in institution names. Techniques like schema matching, record linkage, and data fusion are employed to ensure a coherent integration of heterogeneous data sources (Dong, Halevy, and Yu, 2009). Modern data integration frameworks leverage machine learning to automate and optimize the schema matching and data fusion processes, enhancing the accuracy and scalability of integration tasks (Dong et al., 2022).

2.4.3 Entity Resolution

Entity resolution is the process of identifying and merging records that refer to the same real-world entity across different datasets. In academic knowledge graphs, this involves resolving ambiguities in author names, paper titles, and institution names. Effective entity resolution is crucial for creating accurate and reliable knowledge graphs, as it ensures that each entity is uniquely and correctly represented. Techniques for entity resolution include rule-based approaches, machine learning methods, and probabilistic models. For example, machine learning models can be trained to predict whether two records refer to the same entity based on features like name similarity, co-authorship patterns, and publication venues (Bhattacharya and Getoor, 2007). Advances in deep learning have further enhanced the precision and scalability of entity resolution methods (Mudgal et al., 2018).

2.4.4 Graph Construction

Graph construction involves representing the integrated data as a graph, with nodes for entities and edges for relationships. This step requires defining the schema and ontology for the academic domain. Tools like RDF (Resource Description Framework) and graph databases (such as Neo4j) are often used to store and query the knowledge graph (Angles et al., 2017). The schema may include detailed definitions for each type of entity and relationship, enabling sophisticated queries and analyses. Current trends include the use of property graphs and labeled property graphs that allow richer data representation and support for complex queries (Robinson, Webber, and Eifrem, 2015).

2.4.5 Graph Enrichment

Graph enrichment enhances the knowledge graph with additional information, such as keywords, abstracts, and citation contexts. This enrichment process provides more context and supports advanced queries and analyses. Techniques like natural language processing (NLP) and machine learning are employed to extract and link this additional information to the relevant entities and relationships (Ji et al., 2021). For example, NLP techniques can be used to extract research topics from paper abstracts, linking them to the corresponding paper nodes in the graph. The

advent of transformer-based models such as BERT and GPT has significantly improved the ability to understand and annotate textual data, making the enrichment process more accurate and comprehensive (Devlin et al., 2018).

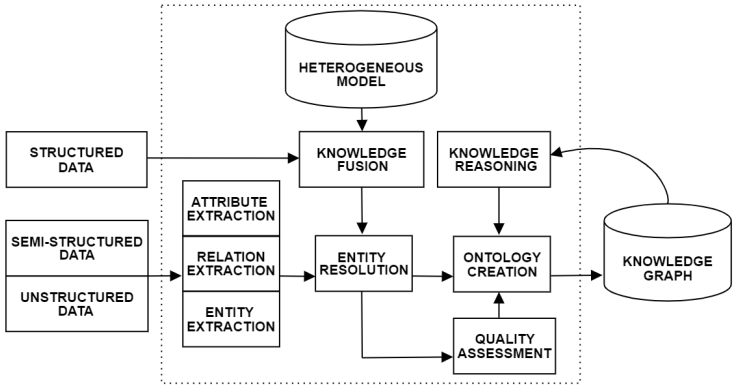

FIGURE 2.2: Academic Knowledge Graph Creation Process. The figure shows the steps involved in constructing an academic knowledge graph depending on the type of input data. (Dong et al., 2022).

2.5 Entity Resolution

TO return to the concept of entity resolution, also known as record linkage or deduplication, this is the process of identifying and linking records that refer to the same real-world entity across different datasets. In the context of academic knowledge graphs, entity resolution involves matching and merging entities such as authors, papers, and institutions to create a coherent and complete graph. Techniques for entity resolution include rule-based approaches, machine learning methods, and probabilistic models.

Rule-based approaches use predefined rules to match entities based on attributes such as names, titles, and affiliations. For example, a rule might state that two author records with the same name and affiliation are likely the same person. These approaches are straightforward but can be limited by their rigidity and inability to handle complex matching scenarios (Christen, 2011). Machine learning approaches involve training models to predict whether two records refer to the same entity based on various features. Supervised learning methods require labeled training data, while unsupervised methods do not. Features used can include textual similarity, co-authorship networks, citation patterns, and publication venues (Bhattacharya and Getoor, 2007). State-of-the-art machine learning techniques leverage neural networks and deep learning models to capture more complex patterns and improve matching accuracy (Mudgal et al., 2018). Probabilistic approaches estimate the likelihood that two records are the same entity using probabilistic models. Techniques like Expectation-Maximization (EM) and Bayesian networks are commonly used in this context. Probabilistic approaches can handle uncertainty and provide a flexible framework for entity resolution (Bhattacharya and Getoor, 2007). Advances in probabilistic graphical models and Bayesian inference have further enhanced the robustness and scalability of these approaches (Li et al., 2020).

2.6 Machine Learning Techniques for Entity Resolution

Machine learning techniques play a significant role in entity resolution, leveraging large datasets and complex features to improve accuracy. Common techniques include supervised learning, unsupervised learning, semi-supervised learning, and deep learning.

Supervised learning models are trained on labeled data where entities have been manually matched. Features may include textual similarity, co-authorship networks, and citation patterns. Examples include decision trees, support vector machines (SVMs), and neural networks. Supervised learning can achieve high accuracy but requires a substantial amount of labeled data. Recent advancements in deep learning, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have further improved the performance of supervised learning models in entity resolution tasks (Li et al., 2020).

Unsupervised learning models learn to identify matching entities without labeled data. Clustering algorithms and density-based methods are commonly used. Unsupervised learning can be useful when labeled data is scarce but may not achieve the same level of accuracy as supervised learning. Techniques such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and hierarchical clustering have been widely applied in entity resolution (Christen, 2011).

Semi-supervised learning combines both labeled and unlabeled data to improve model performance. This approach leverages the availability of a small amount of labeled data along with a large amount of unlabeled data, using techniques like co-training, self-training, and multi-view learning (Bhattacharya and Getoor, 2007). Semi-supervised learning has shown promise in improving the scalability and accuracy of entity resolution models (Li et al., 2020).

Deep learning utilizes neural networks to capture complex patterns in data. Architectures such as CNNs, RNNs, and transformers have been applied to entity resolution tasks with significant success. Deep learning models can automatically extract relevant features from raw data, reducing the need for manual feature engineering. Techniques like Siamese networks and attention mechanisms have further enhanced the performance of deep learning models in entity resolution (Mudgal et al., 2018).

2.7 Graph-Based Techniques for Entity Resolution

Graph-based techniques for entity resolution leverage the structure of the knowledge graph to improve matching accuracy.

Key techniques include graph clustering, link prediction, and graph embeddings. Graph clustering groups entities based on their connections in the graph. Algorithms such as Louvain (Blondel et al., 2008) and Girvan-Newman (Girvan and Newman, 2002) detect communities of related entities. Graph clustering can reveal hidden relationships and improve entity resolution by considering the broader context of each entity (Zhao et al., 2018). Link prediction predicts the likelihood of a relationship between two entities based on existing connections. Techniques such as Adamic-Adar (Adamic and Adar, 2003), Jaccard Coefficient (Jaccard, 1901), and

preferential attachment (Barabási and Albert, 1999) are commonly used. Link prediction can help in identifying missing links and improving the completeness of the knowledge graph (Liben-Nowell and Kleinberg, 2003). Recent advancements in graph neural networks (GNNs) have further enhanced the performance of link prediction models by capturing complex patterns in graph data (Zhang and Chen, 2018). Graph embeddings map entities and relationships to a continuous vector space, preserving graph structure. Techniques like DeepWalk (Perozzi, Al-Rfou, and Skiena, 2014), node2vec (Grover and Leskovec, 2016), and graph convolutional networks (GCNs) (Kipf and Welling, 2016) are commonly used. Graph embeddings can capture the semantic similarity between entities, aiding in entity resolution tasks. The use of graph embeddings has shown significant improvements in scalability and accuracy for large-scale knowledge graphs (Perozzi, Al-Rfou, and Skiena, 2014).

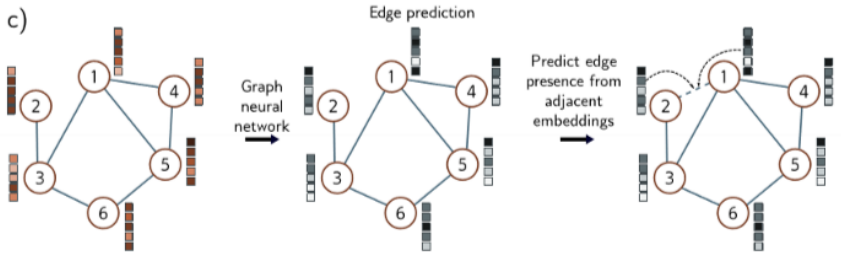

FIGURE 2.3: Link Prediction in Knowledge Graphs. The figure shows how graph neural networks are used to obtain edge predictions from nodes and features (Prince, 2023, CC BY-SA 4.0.).

In summary, the combination of machine learning and graph-based techniques provides powerful tools for entity resolution in academic knowledge graphs. These techniques enhance the accuracy and completeness of knowledge graphs, supporting advanced bibliometric analyses and knowledge discovery. As the field progresses, further integration of these approaches with emerging technologies such as deep learning and graph neural networks promises to push the boundaries of what can be achieved in entity resolution and knowledge graph construction.

Chapter 3

Data

3.1 Introduction

The dataset utilized in this research is sourced from the American Physical Society, encompassing an extensive collection of scholarly articles in the field of physics. This dataset is an invaluable resource for research across various domains, including condensed matter physics, particle physics, and interdisciplinary studies (Society, 2023).

3.2 Properties

The APS dataset is a crucial resource for research and analysis, distinguished by several key attributes. Firstly, its size is substantial, comprising 720,535 rows of metadata that provide information about scholarly articles. When expanded to represent each paper-author pair individually, the dataset encompasses 2,571,821 rows, illustrating its extensive coverage.

Another significant feature of the APS dataset is its diversity. It includes 693,706 citing papers and 625,223 cited papers, reflecting a broad spectrum of scholarly interactions and citations within the field of physics. This diversity enables researchers to explore a wide array of connections and influences between different works, offering rich insights into the academic landscape.

The dataset's detail is also noteworthy. It contains extensive metadata, such as article titles, author names, author affiliations, publication dates, etc. This wealth of information enhances the potential for detailed analysis, allowing researchers to investigate various facets of scholarly articles and their interconnections thoroughly.

Moreover, the APS dataset boasts impressive temporal coverage, spanning from 1893 to 2022. This extensive range offers a longitudinal view of scientific progress in physics over more than a century. Researchers can examine trends, developments, and shifts in the field over this substantial period, gaining valuable historical perspectives.

3.3 Components

The APS dataset is divided into two main components: the citation network and the metadata. Each component contains distinct but interrelated features that are critical for various analytical tasks.

3.3.1 Citation Network

The citation network data is structured to reflect the relationships between citing and cited articles, forming a database of scholarly communication. Only relationships contained within the American Physcial Society are represented.

citing_doi	cited_doi
10.1103/PhysRevSeriesI.11.215	10.1103/PhysRevSeriesI.1.1
10.1103/PhysRevSeriesI.12.121	10.1103/PhysRevSeriesI.1.166
10.1103/PhysRevSeriesI.7.93	10.1103/PhysRevSeriesI.1.166
10.1103/PhysRevSeriesI.16.267	10.1103/PhysRevSeriesI.2.35
10.1103/PhysRevSeriesI.17.65	10.1103/PhysRevSeriesI.2.112

TABLE 3.1: Sample Citation Pairs

Table 3.1 shows an example of citation pairs, illustrating the structure and format of the DOI (Digital Object Identifier) entries in the dataset.

3.3.2 Metadata

The metadata provides detailed information about the articles and their authors, including names and affiliations. This metadata is crucial for author-level analysis and understanding collaboration patterns within the scientific community.

Feature	Description
ID	Unique identifier for the article
Title	Title of the scholarly article
Publisher	Name of the publisher
Journal	Journal in which the article was published
Issue	Issue number of the journal
Volume	Volume number of the journal
Page Start	Starting page number of the article
Page End	Ending page number of the article
Sequence Number	Sequence number of the article
Date	Date when the article was published
Number of Pages	Total number of pages in the article
Article Type	Type of the article (e.g., research article, review)
Identifiers	Unique identifiers for the article (e.g., DOI)
Rights	Rights associated with the article
Authors	Names of authors contributing to the article
Affiliations	Institutions to which authors are affiliated
Has Article ID	Indicates whether the article has an ID
Classification Schemes	Classification schemes associated with the article

TABLE 3.2: Features of the Metadata

Table 3.2 lists the principal features of the metadata dataset, each providing specific information about the articles and their authors.

3.4 Exploratory Data Analysis

An initial exploratory analysis was conducted to gain familiarity with the data before proceeding with more complex analyses.

3.4.1 Citation Network

The citation data represents a complex network of scholarly communication within the APS dataset with a total of 9,883,191 citations. Below is a summary of basic statistics:

	citing_doi	**cited_doi**
Unique Count	693,706	625,223
Most Frequent	10.1103/RevModPhys.83.471	10.1103/PhysRevLett.77.3865
Mean	14.17	15.73
Standard Deviation	12.69	53.70
Maximum	607	14,357

TABLE 3.3: Statistics of the Citation Network

Table 3.3 provides a summary of the basic statistics of the citation data, offering insights into its structure and contents.

The citation distribution statistics indicate a mean of 14.17 citations per citing DOI and 15.73 citations per cited DOI, with standard deviations of 12.69 and 53.70, respectively. This variability underscores the diverse impact and dissemination of research within the dataset.

The top citing papers predominantly originate from journals specializing in modern physics reviews, aligning with the dataset's focus on contemporary scholarly discourse. The most frequently cited paper on gradient descent highlights its broad applicability within physics, underscoring its significant impact and relevance in the field.

As a note, the full set of cited DOIs is contained within the set of citing DOIs. Therefore, when an exhaustive list of papers is needed in further analysis, solely the list of citing DOIs are referenced.

3.4.2 Metadata

Metadata provides essential contextual information about each entry, such as publication years, affiliations, and geographical locations. Understanding these metadata elements is crucial for gaining insights into the dataset's temporal trends, institutional collaborations, and global research contributions. The APS metadata is comprised of articles belonging to the journals shown in Table 3.4.

Table 3.4 shows Physical Review B has the most contributions with 207,609, followed by Physical Review Letters with 133,694 and Physical Review D with 103,017.

Once having gathered the articles from all journals, further analysis reveals that the years 2020, 2021, and 2022 have the highest number of entries, indicating peak publication years within the dataset. Specifically, 2020 saw 98,607 entries, 2021 had 92,835 entries, and 2022 had 90,092 entries. These peaks suggest a recent surge in publication activity, reflecting heightened research efforts during these years.

The distribution of entries per year provides additional context on the dataset's publication frequency. On average, there are approximately 19,453 entries per year,

Journal ID	Journal Name	Entry Count
PRB	Physical Review B	207,609
PRL	Physical Review Letters	133,694
PRD	Physical Review D	103,017
PRA	Physical Review A	88,357
PRE	Physical Review E	65,626
PR	Physical Review	47,940
PRC	Physical Review C	43,842
PRAPPLIED	Physical Review Applied	5,295
PRRESEARCH	Physical Review Research	4,8 13
PRMATERIALS	Physical Review Materials	3,951
PRFLUIDS	Physical Review Fluids	3,524
RMP	Reviews of Modern Physics	3,494
PRSTAB	Special Topics - Accelerators and Beams	2,399
PRX	Physical Review X	2,352
PRAB	Physical Review Accelerators and Beams	1,566
PRI	Physical Review (Series I)	1,469
PRPER	Physical Review Physics Education Research	693
PRXQUANTUM	PRX Quantum	497
PRSTPER	Special Topics - Physics Education Research	368
PRXENERGY	PRX Energy	29

TABLE 3.4: Top Journals by Entry Count

with a notable standard deviation of 28,193 entries. This high variability highlights significant fluctuations in publication volume across different years, with some years seeing substantially more activity than others.

The dataset encompasses a diverse array of affiliations, with a total of 1,292 unique affiliations identified. The top affiliations by entry count include the *Centro de Física de Materiales CSIC/UPV-EHU-Materials Physics Center* in San Sebastián, Spain, which contributes a remarkable 1,468,909 entries. This institution is followed by the *Instituto de Nanociencia y Materiales de Aragón (INMA), CSIC-Universidad de Zaragoza* in Zaragoza, Spain, with 392,092 entries. Lastly, the *Physics Department, Faculty of Science at Al-Azhar University* in Nasr City, Cairo, Egypt accounts for 171,944 entries.

The distribution of entries per affiliation provides insights into the publication output from different institutions. On average, there are 1,939 entries per affiliation, but the standard deviation is substantial at 42,812 entries, indicating significant differences in research output across affiliations. This suggests that while some institutions are highly prolific, others contribute fewer publications. Moreover, 53,731 entries lack affiliation data.

It is important to note that an author can have multiple affiliations, leading to the country variable being represented as a list containing all countries associated with that author. As a result, some entries are counted under multiple countries if the authors are affiliated with institutions in different nations.

The dataset spans 196 unique country sets, with notable contributions from the combination of Germany and Spain, Spain individually, and Egypt. Specifically, the combination of Germany and Spain accounts for 1,641,368 entries, Spain alone has

472,729 entries, and Egypt contributes 172,163 entries. However, there is a notable portion of the dataset, comprising 60,450 entries, that lacks specific country information.

The distribution of entries per country highlights varying levels of research contributions globally. The mean number of entries per country is 12,903, with a wide standard deviation of 122,458, indicating disparities in research output among different nations. The maximum number of entries from a single country combination (Germany and Spain) is 1,641,368, underscoring the collaborative research strength of these countries.

In conclusion, this section provides a detailed analysis of the metadata extracted from the APS dataset, offering insights into publication trends, affiliation distributions, and global contributions. Understanding these characteristics is crucial for further exploration and analysis of the dataset. By examining these metadata elements, we better understand the temporal, institutional, and geographical dimensions of research activity in the dataset.

3.5 Creation of Data Tables

With the data now imported and validated, several data tables were created with separate focuses to aid in ease of interpretation and flexibility of use for further analyses.

3.5.1 Author Table

To analyze author contributions, author information was extracted, parsed from the metadata, and stored in an **author_table**.

Field	Details
Identifier	0
DOI	10.1103/PhysRev.1.124
Type	Person
Name	Lachlan Gilchrist
First Name	Lachlan
Surname	Gilchrist
Year	1913
Affiliation	Centro de Física de Materiales, San Sebastián, Spain Donostia International Physics Center, San Sebastián, Spain Physics Department, Technical University of Munich, Garching, Germany
Countries	['Germany', 'Spain']

TABLE 3.5: First Entry of the Author Table

Table 3.5 illustrates the structure of the **author_table** by displaying the details of the first entry.

The "Type" column in the author table differentiates between individual authors and collaborative groups. There are two possible values in this column: "Person" (2,570,491 instances) and "Collaboration" (1,330 instances). All records corresponding to collaboration were disregarded for this work as the focus remained on the contributions and disambiguation of individuals.

3.5.2 Paper Table

A separate table, referred to as the **paper_table**, was created to uniquely identify each paper based on its DOI. The structure of the paper table is presented in Table 3.6.

Identifier	DOI
0	10.1103/PhysRevSeriesI.11.215
1	10.1103/PhysRevSeriesI.12.121
2	10.1103/PhysRevSeriesI.7.93
...	...
709802	10.1103/PhysRevFluids.7.093104

TABLE 3.6: Structure of the Paper Table

3.5.3 Merged Table

Finally, the **merged_table** used for analyses combines metadata and citation data, linking papers by their DOI. This was achieved through a simple join operation. The structure of the merged table is identical to the **author_table** with the addition of the paper identifier value. In order to avoid overlap between author and paper identifiers, paper identifier values are shifted to start following the maximum of the author identifier values.

3.6 Data Quality

Unlike many open-source databases, the citation network and metadata of our dataset have been meticulously curated by journal editors and subject matter experts. This curation ensures higher data quality, accuracy, and completeness, which are critical for reliable scientific analysis and research. By leveraging curated data, potential issues such as incomplete records or misclassified information are minimized, providing a more robust foundation for scholarly investigations. Nevertheless, analyses are still constrained by the quality of the dataset itself. These limitations arise due to the nature of data collection, potential inaccuracies, and the completeness of the dataset.

3.7 Summary

This chapter provides a detailed overview of the APS dataset, highlighting its key characteristics essential for subsequent analyses. It details the exploratory data analysis undertaken to become familiarized with the dataset and the creation of structured data tables such as the **author_table**, **paper_table**, and **merged_table**. The chapter concludes by recognizing both the data quality and limitations.

Chapter 4

Author Grouping Strategies & Citational Overlap

4.1 Introduction

In this chapter, we perform analyses to understand the relationships between papers and their authors, providing a foundation for graph-based completion and entity resolution. We focus on pairwise comparisons of papers for each author, author cosine similarities, and various author grouping strategies. To manage computational constraints and maintain a manageable dataset size, analyses are performed only on papers published in *Physical Review E*. This journal was chosen because it provides a moderately sized dataset that is sufficient for our analysis while being feasible to handle within our computational resources, allowing us to focus on detailed modeling without being overwhelmed by data volume.

4.2 Analyses for Full-Name Matching

In our approach, understanding the citation habits of authors is crucial for entity resolution and graph completion in academic knowledge graphs. Citation patterns can reveal relationships between papers, indicating whether they are authored by the same individual or authors working within similar research domains. To achieve this, we need more than one paper per author to analyze citational overlap effectively. We leverage methods from existing research, such as Schulz et al., 2014, which demonstrates the utility of citation networks in large-scale author name disambiguation.

4.2.1 Pairwise Comparisons

We begin by grouping authors based on their full-names, and we filter out groups with fewer than two DOIs. For each group of authors, we retrieve the citations associated with their DOIs from the citation dataset.

To facilitate pairwise comparison, we create binary matrices for each author, where the rows represent the citing DOIs and the columns represent the cited DOIs. Each entry in the matrix is binary, indicating whether a citation exists between the corresponding papers. This structured representation allows us to compute various statistical measures that highlight patterns and relationships within the citation network.

4.2.2 Cosine Similarities

To deepen our analysis, we use cosine similarity to compute correlation matrices for the binary citation matrices. Cosine similarity is useful for high-dimensional sparse data, measuring the cosine of the angle between two non-zero vectors. The formula for cosine similarity between two vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ is given by:

$$\text{cosine\_similarity}(\mathbf{v_1}, \mathbf{v_2}) = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \|\mathbf{v_2}\|}$$

Multiprocessing is employed to parallelize this computation, reducing the time required to process large datasets (Dean and Ghemawat, 2008). The resulting correlation matrices are analyzed to extract meaningful patterns in citation behavior.

By examining the distribution of cosine similarities, we identify the average citation overlap between authors' papers. This measure helps in understanding the extent of self-citation or citation of closely related work, crucial for entity resolution tasks where distinguishing between closely related entities is challenging (Hirsch, 2005).

The plot in Figure A.1 in the appendix displays the distribution without including authors who possess zero citational overlap. Intuitively, this may occur when an author has not written many papers, meaning the opportunity to cite the same work(s) is much lower. In order to check this, we analyzed the distribution of the number of papers for authors with minimal citational overlap (Figure A.2 in the appendix). Our intuitions are confirmed by our findings.

4.3 Additional Author Grouping Strategies

Author grouping is a critical step in ensuring accurate citation and authorship linkages. The analyses presented above were all performed by matching the full-name of an entity. We went on to explore two additional author grouping strategies for comparison: surname matching and normalized indel distance matching.

4.3.1 Surname Matching

The analyses for surname matching are identical to those performed on full-name matching. However, rather than matching authors by their full-name before performing analyses, authors are matched solely by their surname. With this relaxed requirement, we have a smaller number of authors to analyze as more entities are merged. This, however, also results in a larger amount of citations belonging to each entity. Additionally, we perform the same check to ensure authors with minimal citational overlap have written few papers. The results are shown in Figure A.3 and Figure A.4 in the appendix.

4.3.2 Normalized Indel Distance Matching

For our final author grouping strategy, we apply normalized indel distance, a string similarity measure. Using the `rapidfuzz` package (Bachmann, 2024), we set a distance threshold of 80 out of 100 to identify and merge author names with minor

discrepancies. This method is employed with the intention of mitigating issues arising from typographical errors or naming variations.

Normalized indel distance measures the number of insertions or deletions required to change one string into another, normalized by the length of the strings (Cohen, Ravikumar, and Fienberg, 2003). Our implementation with `rapidfuzz` efficiently computes this distance, allowing us to merge DOIs for similar names (Cohen, Ravikumar, and Fienberg, 2003).

Authors are first grouped based on their full names, with their associated DOIs identified and recorded. The `rapidfuzz` library is then used to find author names that are similar, detecting minor variations and typographical errors. DOIs for authors whose names exceed the predefined similarity threshold are merged to consolidate publications originally attributed to different variations of the same author's name. Throughout this process, records of processed authors are maintained to avoid redundant computations and ensure efficiency. Author names merged by normalized indel distance are standardized based on the first occurrence in the iteration of matching. The results for author grouping based on normalized indel distance matching are shown in Figures A.5 and A.6 in the appendix.

To further understand the impact of author grouping strategies, we analyze the frequency distributions of DOIs per author using a log-log plot. This provides insight into the distribution patterns and highlights differences between grouping methods.

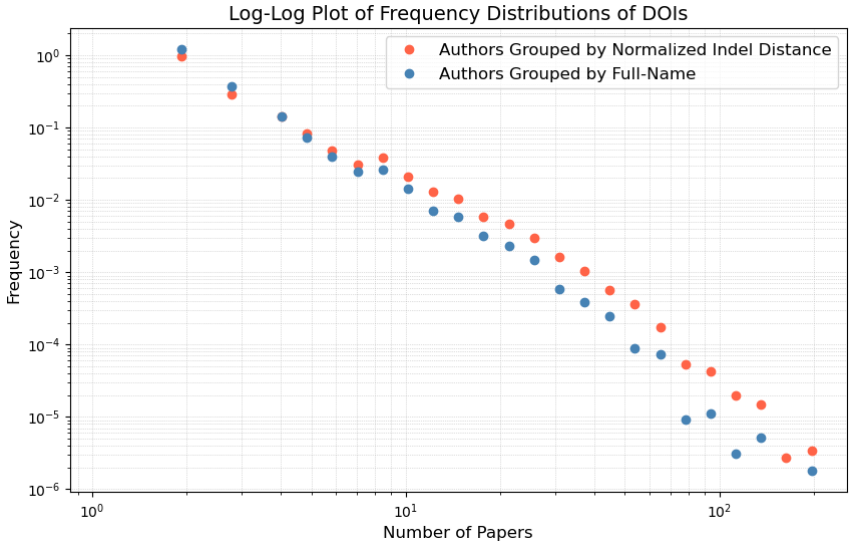

FIGURE 4.1: Log-Log Plot of Frequency Distributions of DOIs for Authors Grouped by Full-Name and Normalized Indel Distance

From Figure 4.1, we observe that the frequency distribution resembles a power-law pattern, which is common in many natural and social phenomena. The distribution for normalized indel distance matching (red points) and full-name matching (blue points) shows a similar trend, with deviations at higher numbers of papers. This deviation indicates that normalized indel distance matching results in a higher frequency of authors with a larger number of papers, reflecting the merging of more author entities compared to full-name matching.

4.4 Comparison of Author Grouping Strategies

We compare the original author groupings with those obtained through surname matching and normalized indel distance matching. Key statistics include the number of authors, mean number of papers per author, and the standard deviation of papers per author. These statistics help assess the impact of name standardization on the dataset's structure.

Grouping Method	Author Count	Mean Papers	STD Papers
Full-Name	30,659	4.65	5.93
Surname	20,582	8.21	26.80
Normalized Indel Distance	24,390	6.55	9.13

TABLE 4.1: Comparison of Author Grouping Strategies

From Table 4.1, we observe that the full-name grouping method results in the highest number of unique authors, indicating a more granular approach. In contrast, surname matching significantly reduces the author count while increasing the mean and standard deviation of papers per author, reflecting the merging of more entities under common surnames. The normalized indel distance method strikes a balance between the two, offering moderate author count and paper metrics, which suggests effective mitigation of minor discrepancies in author names. These variations highlight the importance of selecting a suitable author grouping strategy based on the specific requirements of the analysis.

Finally, we construct a graph to perform a side-by-side comparison of the distribution of mean author cosine similarity for each grouping method to understand the extent of citation overlap among authors. This analysis helps reveal how relaxing the matching criteria affects the citation patterns.

Higher mean cosine similarity indicates greater citation overlap, suggesting stronger connections between papers authored by the same person or entity. Lower mean cosine similarity implies more diverse citation patterns, which occurs due to less precise author grouping. Our analyses reveal that full-name matching achieved a mean author cosine similarity of 0.1557, surname matching 0.1160, and normalized indel distance matching 0.1404.

4.5 Conclusion

In conclusion, our analyses reveal that different author grouping strategies significantly impact the structure of the dataset and the inferred citation patterns. Full-name matching, with a mean cosine similarity of 0.1557, provides a granular view with higher author counts and lower mean papers per author, indicating strong connections and precise author grouping. This suggests higher citational overlap and effective identification of individual authors. Surname matching, with a mean cosine similarity of 0.1160, results in the highest author counts and the largest number of papers per author, reflecting more diverse citation patterns due to relaxed matching criteria, leading to lower citational overlap and potential conflation of different authors. Normalized indel distance matching, with a mean cosine similarity of 0.1404, balances precision and inclusiveness, effectively mitigating minor discrepancies in

FIGURE 4.2: Histogram Comparison of Mean Author Cosine Similarities for Different Author Grouping Strategies

author names. This approach provides moderate citational overlap, making it useful for entity resolution where minor name variations occur. The choice of author grouping strategy should be informed by the specific goals and requirements of the analysis at hand. In subsequent analyses, full-name matching is used as the author grouping strategy, as it is assumed to be the ground truth due to its higher precision in identifying individual authors. We assume the dataset from APS to be of high quality and ignore potential anomalies in the data.

Chapter 5

Graph Neural Network

5.1 Introduction

In this chapter, the design, training, and evaluation of a Graph Neural Network (GNN) for link prediction in an academic knowledge graph are addressed. The process begins with the construction of the graph from tabular data, where nodes and edges are defined to represent authors, papers, and their relationships. Subsequently, the edge-level splits for creating training, validation, and test sets are explained. Mini-batch loading is then covered to optimize the training process for large-scale heterogeneous graphs. The GNN model architecture is detailed, including the use of graph convolution layers and the classifier for link prediction. The chapter concludes with a thorough evaluation of the model's performance using metrics such as AUC, precision, recall, and F1-score. Note that the dataset remains focused exclusively on papers from the *Physical Review E* journal.

5.2 Foundation

Graph Neural Networks (GNNs) are a class of deep learning models designed to handle graph-structured data. Unlike traditional neural networks that operate on fixed-size input vectors, GNNs can process data where relationships between entities are represented as edges connecting nodes in a graph. This ability makes GNNs particularly suitable for tasks involving relational data, such as social networks, molecular structures, and knowledge graphs (Prince, 2023).

The fundamental operation of a GNN involves iteratively updating the representation of each node by aggregating information from its neighbors. This process, often referred to as message passing or graph convolution, allows nodes to learn embeddings that capture both their features and the structure of the graph around them (Kipf and Welling, 2016; Gilmer et al., 2017). Typically, a GNN consists of multiple layers of graph convolution operations, each layer enabling nodes to incorporate information from progressively larger neighborhoods (Wu et al., 2020).

In each layer, the node features are updated based on a combination of their own features and the features of their neighboring nodes. This is achieved through operations such as weighted sums, followed by the application of non-linear activation functions (Hamilton, Ying, and Leskovec, 2017). By stacking multiple layers, GNNs can learn complex patterns and dependencies in the graph, making them powerful tools for a wide range of applications, including node classification, link prediction, and graph classification (Zhou et al., 2020).

5.3 Graph Construction

The first step required is the construction of a knowledge graph from our tabular data. This means integrating information about authors, papers, authorship edges, and citation edges. The creation of this graph involves several key steps, including mapping identifiers, assigning numeric codes, constructing edge indices, converting data to tensors, and creating a heterogeneous data object in preparation for our model.

5.3.1 Node Definition

The first step involves mapping author identifiers and paper DOIs to unique numerical IDs. The 'author_identifier' column is initially converted to a categorical data type, allowing each unique author to be mapped to a corresponding numerical ID. For the papers, a mapping is created for the DOIs by concatenating the citing and cited from the citation dataset, keeping unique values, and then enumerating them. This process ensures that each author and paper has a unique numerical representation, which is crucial for subsequent operations. These mappings serve as our nodes for authors and papers.

5.3.2 Feature Extraction

In addition to the basic identifiers, features for authors and papers were included to enhance the model's capability. Each author is associated with their respective countries and a set of derived features. The 'countries' feature for authors was one-hot encoded from a list of fifteen unique countries. This encoding ensures that the multi-national nature of an author's affiliations is captured accurately. For papers, the year of publication was normalized and included as a feature. These features provide additional contextual information that aids in more accurate link prediction.

To further augment the model's predictive power, a vector of ones of dimension 7 was also introduced as an input feature. This vector acts as a constant bias term, ensuring that the model can effectively capture the baseline conditions or intercepts during training. Including this feature alongside the categorical and derived features maintains consistency across all input data points and prevents any bias from affecting the overall performance of the model.

5.3.3 Edge Construction

The connected structure of the graph is defined by its edges, which represent relationships between nodes. Two types of edge indices are initialized: citations between papers and authorship relationships between authors and papers. The citation edge index is populated by iterating over the citation dataset. For each citation, the citing and cited DOIs are mapped to their corresponding numerical IDs based on the previously created mapping. The authorship edge index is constructed similarly. Each paper's DOI and author identifier are mapped to their numerical IDs. During this process, any missing papers or authors are tracked to ensure the integrity of the mappings. Finally, to facilitate efficient computation, the edge indices are converted to PyTorch tensors. This conversion ensures that the data is stored in a format suitable for high-performance computing, enabling the use of advanced machine learning algorithms. The tensors are also stored in contiguous memory, which is a best practice for optimizing the performance of tensor operations.

5.3.4 Heterogeneous Data Object Creation

Using the `HeteroData` class from PyTorch Geometric, a heterogeneous graph is created (Fey and Lenssen, 2022a). This class allows for the definition of nodes and edges with different types, accommodating the complexity of the academic network. Two types of nodes are defined: `paper` and `author`, along with three types of edges: `cites` (from paper to paper), `written_by` (from paper to author), and `writes` (from author to paper). This structure effectively models the multi-relational nature of the data.

The final graph consists of 139,966 'paper' nodes and 192,814 'author' nodes. Paper features include eight attributes, while author features include 22 attributes. The citation edge index contains 700,250 edges, representing citation relationships between papers. The authorship edge index contains 192,392 edges, representing authorship relationships between authors and papers, with an additional reverse authorship edge index containing 192,392 edges to maintain bidirectional relationships in the graph.

Statistic	**Value**
Number of 'paper' Nodes	139,966
Number of 'author' Nodes	192,814
Paper Feature Shape	(139966, 8)
Author Feature Shape	(192814, 22)
Citation Edge Index Shape	(2, 700,250)
Authorship Edge Index Shape	(2, 192,392)
Reverse Authorship Edge Index Shape	(2, 192,392)

TABLE 5.1: Node, Feature, & Edge Statistics

5.4 Edge-Level Splits

In the context of training graph neural networks (GNNs) on large-scale heterogeneous graphs, it is crucial to manage data effectively to ensure efficient learning and evaluation. This process involves splitting the dataset into training, validation, and test sets using edge-level splits. The specific method employed for this task is the `RandomLinkSplit` transform from the PyTorch Geometric library, which allows for full data partitioning while maintaining a balanced distribution of edges and nodes across the subsets (Fey and Lenssen, 2022b).

5.4.1 Transformation

The `RandomLinkSplit` transform is used to divide the dataset into training, validation, and test sets with predefined ratios for validation (10%) and test (10%) edges. This transformation method also incorporates a disjoint training ratio (0.30) and negative sampling ratio (2.0), enhancing the model's learning capability by providing a mix of positive and negative examples. The disjoint training ratio ensures that a certain portion of the edges used in the training set does not overlap with those in the validation or test sets, reducing the risk of overfitting and improving generalization (Kipf and Welling, 2016). The negative sampling ratio controls the proportion of negative (non-existing) edges to positive (existing) edges in the training set, ensuring that the model learns to distinguish between real and spurious connections effectively.

5.4.2 Configuration

The application of the `RandomLinkSplit` transform results in three distinct data subsets: training, validation, and testing sets. The training set contains a specified number of positive edges, representing actual relationships. Notably, this subset does not include negative samples, allowing the model to focus on learning the inherent patterns of positive relationships without distraction (Hamilton, Ying, and Leskovec, 2017).

For the validation and test sets, the transform introduces negative sampling. Negative samples are synthetic edges that do not exist in the original graph, simulating the absence of a relationship between nodes. This approach creates a balanced ratio of negative to positive samples, aiding the model in learning to distinguish between actual and non-existent relationships (Fey and Lenssen, 2019). The validation set is used to tune hyperparameters and monitor the model's performance during training, while the test set is reserved for the final evaluation.

5.5 Mini-Batch Loading

To handle the complexity and scale of large heterogeneous graphs, the implementation employs mini-batch loading, a technique that allows for efficient and scalable training of graph neural networks (GNNs). The `LinkNeighborLoader` from the PyTorch Geometric library is designed to sample a neighborhood of nodes and edges from the graph to create mini-batches (Fey and Lenssen, 2022c). This loader is particularly useful for link prediction tasks where the goal is to predict the existence of edges between nodes in the graph. The process begins by defining the seed edges, which are the edges from the training set that the loader will sample around.

5.5.1 Theoretical Foundations

The mini-batch loading technique is grounded in the principles of stochastic gradient descent (SGD), which allows for efficient training on large datasets by updating model parameters using small, random subsets of the data (Robbins and Monro, 1951). In the context of GNNs, mini-batch loading enables the model to learn from localized graph structures, preserving the dependencies between nodes while reducing the computational complexity compared to full-batch training. This approach improves the model's robustness and generalization ability, leading to better performance on unseen data (Mikolov et al., 2013).

5.5.2 Practical Implementation

In this implementation, the `LinkNeighborLoader` is configured with the following parameters: the number of neighbors to sample at each layer (20 neighbors at the first layer and 10 at the second layer), a negative sampling ratio of 2.0 (meaning two negative samples for every positive sample), and a batch size of 128. These settings ensure that each mini-batch is representative of the graph's structure while maintaining manageable computational loads (Hamilton, Ying, and Leskovec, 2017). In order to ensure successful implementation, the edge label index size in the sampled data is checked to confirm that it matches the expected size of three times the batch size, accounting for the negative sampling ratio.

5.6 Heterogeneous Link-Level GNN Model Definition

The implementation of the heterogeneous link-level Graph Neural Network (GNN) model is designed to capture complex relationships in the academic knowledge graph, focusing on the interactions between papers and authors. This section details the definition of the GNN model, the classifier, and the overall model architecture.

5.6.1 Model Architecture

The GNN model is constructed using the `SAGEConv` layers from the PyTorch Geometric library, specifically leveraging the `SAGEConv` layer for its ability to aggregate information from a node's local neighborhood effectively (Fey and Lenssen, 2022d). The model comprises four `SAGEConv` layers, each followed by a ReLU activation function, except for the final layer. This design allows the network to learn increasingly abstract representations of the nodes at each layer (Hamilton, Ying, and Leskovec, 2017; Fey and Lenssen, 2019).

The choice of four layers is informed by the nature of the task. In the context of link prediction between authors, it is essential to capture multi-hop relationships within the graph. Each `SAGEConv` layer aggregates information from one-hop neighbors, thus four layers allow the model to consider up to four-hop neighborhoods. This depth is particularly useful for tasks that involve predicting connections between entities that are not directly connected but are linked through intermediate nodes, such as co-authorship through multiple papers.

A two-hop model can effectively capture direct co-authorship relationships, where authors have collaborated directly. However, a four-hop model extends the neighborhood reach, allowing it to capture co-citation relationships as well, where authors are indirectly connected through the papers that cite them. This broader reach is crucial for understanding the more complex structures within an academic knowledge graph.

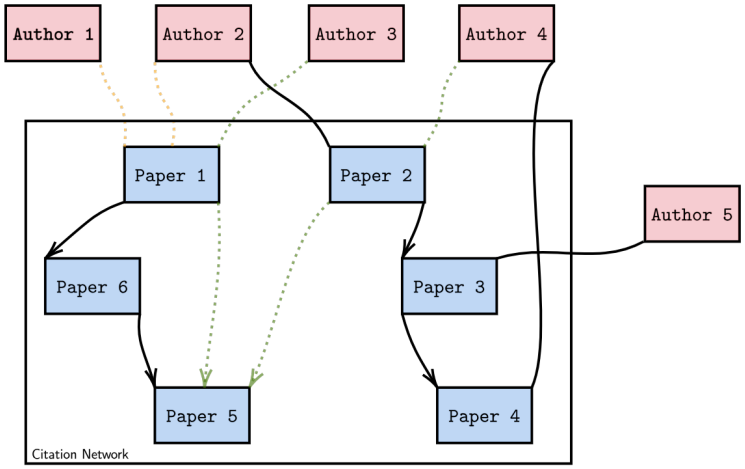

FIGURE 5.1: Graph illustrating author-paper (undirected) and paper-paper (directed) relationships. Orange connectors show co-authorship, while the green connectors show paths to co-citation.

In Figure 5.1, the orange connectors combine to form co-authorship connections which indicate collaboration between authors on a paper. For instance, Author 1 and Author 2 are connected to Paper 1, indicating they co-authored this paper. On the other hand, the green connectors illustrate co-citation paths, which are a combination of authorship and citation edges indicating connectedness of two authors based on their citation habits. For example, Author 4 writes Paper 2, Paper 1 and Paper 2 both cite Paper 5, and Author 3 writes Paper 1. By analyzing this combination of edges through multiple hops, we can see that Author 3 and Author 4 are indirectly connected through the papers that cite their work. By analyzing both types of relations, the model can effectively learn from both co-authorship and co-citation connections, enhancing its predictive capabilities for link prediction tasks.

Testing of models with only 2 or 3 layers revealed significantly higher loss compared to the 4-layer model, indicating that the additional layers and the ability to capture co-citation are indeed beneficial. As seen in the results, the incorporation of co-citation relationships has a substantial impact, demonstrating that the network is effectively leveraging this extended information to improve performance.

In addition to the graph structure, the model incorporates various node features to enhance its predictive capabilities. For papers, the features include normalized publication years, while for authors, one-hot encoded country information is added. These features provide additional contextual information that is crucial for accurate link prediction.

The publication year for each paper is mapped and normalized to fall within a 0-1 range using MinMaxScaler. This normalized year is then concatenated with the initial paper features. For the authors, their country information is one-hot encoded, transforming the categorical country data into a binary vector. Each element of this vector represents the presence or absence of a specific country. This vector is subsequently concatenated with the initial author features.

5.6.2 Classifier Definition

The classifier component of the model is responsible for predicting the existence of edges between nodes, specifically between papers and authors. It does so by computing the dot product of the features of the possibly connected nodes (papers and authors) at the specified edges. This approach effectively captures the interaction between node pairs and is suitable for link prediction tasks (Fey and Lenssen, 2019).

5.6.3 Heterogeneous Transformation

The overall model integrates the GNN with linear transformations for both paper and author features. These transformations map the input features to a hidden space of a dimensionality of 64. The model is then converted into a heterogeneous variant using the `to_hetero` function from PyTorch Geometric, which adapts the GNN to handle the heterogeneous nature of the data, characterized by different node types and edge types (Radicchi, Fortunato, and Vespignani, 2011).

In the forward pass, the model first transforms the paper and author features using linear layers. It then applies the GNN to these features and the edge indices, propagating information through the network. Finally, the classifier predicts the

edges based on the transformed node features, enabling the model to infer relationships between papers and authors.

The choice of hidden channels and the number of layers is crucial for balancing model complexity and computational efficiency. The hidden dimensionality, set to 64 in this implementation, is selected to provide sufficient capacity for learning intricate patterns in the data without overfitting. The number of layers corresponds to the maximum number of hops needed to connect two authors through their authored papers, reflecting the depth of the graph structure considered during training.

After defining the homogeneous GNN, it is converted into a heterogeneous variant using the `to_hetero` function. This function adapts the GNN model to handle multiple types of nodes and edges, creating a `HeteroSAGEConv` layer. The `HeteroSAGEConv` layer extends the SAGEConv mechanism to heterogeneous graphs by managing different types of nodes and edges separately (Dong et al., 2022). This is crucial for datasets where nodes and edges represent different entities and relationships, such as papers and authors in an academic graph.

In the overall model, the paper and author features are first transformed using linear layers to match the input dimensions required by the GNN. These transformed features are then processed through the `HeteroSAGEConv` layers, where the aggregation of neighborhood information occurs separately for each type of node and edge. This allows the model to capture the intricate relationships between different types of entities. Finally, a classifier predicts the edges based on the transformed node features, enabling the model to infer relationships between papers and authors with high accuracy.

To account for the varying importance of different edge types (e.g., citations versus co-authorships), the `HeteroSAGEConv` layers incorporate edge type-specific weights during the aggregation process. This adaptive weighting mechanism allows the model to prioritize and integrate information differently based on the relevance and context of each relationship type within the graph. By distinguishing between edge types, the model enhances its ability to capture nuanced dependencies and improve predictive performance for link prediction tasks in heterogeneous academic knowledge graphs.

In summary, the heterogeneous link-level GNN model is carefully designed to leverage the strengths of the `SAGEConv` layers and the heterogeneous capabilities of PyTorch Geometric, providing a robust framework for link prediction tasks in the academic knowledge graph.

5.7 Training

The training process for the heterogeneous link-level GNN model involves several stages, including setting the model to training mode, iterating through the training data, transferring data to the computational device, computing predictions and losses, performing backpropagation, and optimizing the model parameters. This section outlines these steps in detail.

5.7.1 Process Overview

Each epoch consists of the many activities aimed at optimizing the model's parameters. At the beginning of each epoch, the model is set to training mode using `model.train()`. This enables the model to update its parameters based on the computed gradients (Goodfellow, Bengio, and Courville, 2016). The training data is then iterated through in mini-batches using the configured data loader. This process ensures efficient and manageable computation by breaking down the data into smaller, more manageable subsets (Bottou, 2010). Next, each mini-batch of data is transferred to the computational device (e.g., CPU or GPU). This step ensures that the data is available for computation on the device where the model is being trained (Paszke et al., 2019). For each mini-batch, the model makes predictions, and the loss is computed using a binary cross-entropy loss function. This function is suitable for binary classification tasks, such as link prediction, where the goal is to determine whether a link exists or not (Murphy, 2012). The gradients are computed via backpropagation, and the model parameters are updated using the Adam optimizer, an adaptive learning rate optimization algorithm that combines the benefits of AdaGrad and RMSProp to improve training efficiency and performance (Kingma and Ba, 2014). This process minimizes the loss function, thereby improving the model's performance over successive epochs (Kingma and Ba, 2014).

5.7.2 Per-Epoch Activity

In order to observe per-epoch activity, after each epoch, the model's performance is evaluated on the validation data. An epoch refers to one complete pass through the entire training dataset, and it is a fundamental concept in iterative machine learning processes (Goodfellow, Bengio, and Courville, 2016). This involves setting the model to evaluation mode, computing predictions on the validation mini-batches, and calculating the average validation loss. The evaluation helps monitor the model's generalization capability and guides hyperparameter tuning (Bishop, 2006).

The systematic training and evaluation of the heterogeneous link-level GNN model ensure that it learns effectively from the training data while maintaining high-levels of performance on unseen validation data. This approach leverages advanced techniques in deep learning and graph neural networks, providing a robust framework for predicting relationships in complex knowledge graphs. Four epochs were used after observing the validation loss increase on the fifth epoch when training continued.

Epoch	Training Loss	Validation Loss
001	0.3835	0.2305
002	0.3633	0.2266
003	0.3617	0.2253
004	0.3624	0.2236

TABLE 5.2: Training and Validation Loss per Epoch

Table 5.2 shows the training outcomes achieved using four epochs and binary cross entropy loss.

5.7.3 Trainable Parameters

In a graph neural network (GNN) model, parameters are primarily associated with the linear transformation layers and the convolutional layers that operate on the graph's nodes and edges. Each parameter in these layers has a specific role in transforming the input features into meaningful representations that the model uses to make predictions (Kipf and Welling, 2016).

Trainable parameters are those that the optimization algorithm adjusts during the training process. These include weights and biases in linear layers and convolutional layers. The total number of trainable parameters is a direct indicator of the model's capacity to learn complex patterns from the data. The total number of trainable parameters in the model is 77,408. This count includes weights and biases from both the linear layers and the convolutional layers within the GNN architecture.

5.8 Evaluation

The evaluation of the graph neural network (GNN) model is a crucial step to assess its performance and generalization capability. This section describes the evaluation process, the metrics used to quantify the model's performance, and the results obtained.

5.8.1 Process Overview

Before making predictions, `LinkNeighborLoader` is applied to the test set, and the model is set to evaluation mode using `model.eval()`. This step disables certain layers that behave differently during training and evaluation, such as dropout layers, ensuring that the model's performance is evaluated accurately.

Using the test data loader, the model generates predictions for each mini-batch of test data. The predictions are compared against the ground truth labels to compute various evaluation metrics. The primary task is link prediction, where the model predicts whether a link between two nodes exists (Kipf and Welling, 2016).

5.8.2 Evaluation Metrics

Several metrics are used to evaluate the model's performance, including the Area Under the Receiver Operating Characteristic Curve (AUC), the confusion matrix, and a detailed classification report. These metrics provide an extensive view of the model's accuracy, precision, recall, and F1-score.

The AUC is a widely used metric for binary classification tasks. It measures the model's ability to distinguish between positive and negative classes, with higher values in range [0,1] indicating better performance. The AUC for the test set was computed to be 0.9771, suggesting that the model has a high capability to correctly classify positive and negative links (Fawcett, 2006).

The confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positive, true negative, false positive, and false negative predictions. The confusion matrix for the test set is as follows:

	Predicted Negative	Predicted Positive
Actual Negative	36,628	1,850
Actual Positive	869	18,370

TABLE 5.3: Confusion Matrix for Test Data

Table 5.3 shows that out of 38,478 actual negative examples of author-paper relations, the model correctly identified 36,628 and incorrectly identified 1,850. Similarly, out of 19,239 actual positive examples of author-paper relations, the model correctly identified 18,370 and incorrectly identified 869. These results indicate a high level of accuracy and a good balance between precision and recall (Provost, Fawcett, Kohavi, et al., 1998).

The classification report provides a detailed summary of precision, recall, and F1-score for both classes (negative and positive). Precision is the ratio of true positive predictions to the total number of positive predictions. Recall is the ratio of true positive predictions to the total number of actual positive examples. F-1 score is the harmonic mean of precision and recall, providing a single metric that balances both aspects. The classification report for the test set is as follows:

	Precision	Recall	F1-Score
Negative	0.98	0.95	0.96
Positive	0.91	0.95	0.93

TABLE 5.4: Classification Report for Test Data

These metrics reveal that the model achieves a precision of 0.98 for negative predictions and 0.91 for positive predictions. The recall is 0.95 for negatives and 0.95 for positives, resulting in F1-scores of 0.96 and 0.93, respectively. The overall accuracy of the model is 0.95, indicating robust performance across both classes (Powers, 2020).

5.8.3 Results Discussion

The evaluation results demonstrate that the GNN model performs exceptionally well on the test set, achieving high scores across various metrics. The high AUC value indicates strong discriminative power, while the precision, recall, and F1-scores suggest a good balance between identifying true positives and avoiding false positives. The confusion matrix and classification report further validate the model's effectiveness in predicting both positive and negative links, with a slightly better performance in identifying negative links. This is expected given the higher support (number of instances) for the negative class, which is common in many real-world datasets where non-relationships outnumber relationships (Chawla et al., 2002).

The inclusion of the country and publication year as features did not significantly improve performance, likely due to the relatively young age of the *Physical Review E* journal. Since the journal's publication history aligns well with the careers of its authors, the year feature might not add substantial discriminative value. Nonetheless, the model did show a slight improvement with the addition of these features. To further enhance the model's performance, it is suggested to incorporate affiliations as nodes and countries as features of these nodes. This approach could provide additional context, capturing more nuanced relationships and collaborations within the

academic community.

Overall, the evaluation metrics indicate that the GNN model is well-suited for the link prediction task in the academic knowledge graph, providing reliable and accurate predictions that can be used for further analysis and applications.

5.9 Conclusion

This chapter detailed the construction, training, and evaluation of a Graph Neural Network (GNN) for link prediction in an academic knowledge graph. The GNN, featuring four `SAGEConv` layers, effectively captured multi-hop relationships necessary for predicting links between authors and papers. The evaluation demonstrated the model's robust performance, with an AUC of 0.9771 indicating strong discriminative power. Precision and recall scores were high, with F1-scores of 0.96 for negative links and 0.93 for positive links, confirming the model's accuracy and balance in predictions. The confusion matrix further validated the model's effectiveness in distinguishing between actual and non-existent links. In summary, the GNN model developed in this chapter proved highly effective for link prediction in academic knowledge graphs, delivering accurate and reliable results.

Chapter 6

Conclusion

This thesis has explored the application of graph neural networks (GNNs) for link prediction within an academic knowledge graph, focusing on the intricate relationships between authors and papers. Through a study encompassing data pre-processing, exploratory analysis, and model development, several key findings and contributions have emerged.

The foundation of this thesis rests on the recognition of academic knowledge graphs as complex, interconnected networks that capture the dynamic relationships between entities such as authors and papers (Hamilton, Ying, and Leskovec, 2017). Traditional methods often struggle to effectively model the heterogeneous and multi-relational nature of these graphs. In contrast, GNNs have demonstrated their efficacy in capturing and leveraging the structural dependencies inherent in such data, thereby enhancing link prediction accuracy (Kipf and Welling, 2016).

The initial phase of this study involved the construction of an academic knowledge graph from raw tabular data sourced from the American Physical Society. This process encompassed meticulous data cleaning and entity resolution analysis. By ensuring data integrity and consistency, the resulting graph provided a reliable foundation for subsequent analyses and modeling efforts.

A critical aspect of this thesis was the exploration of author grouping strategies and citation overlap analysis within the knowledge graph. Author grouping revealed distinct clusters of researchers based on full-name, surname, and normalized indel distance matching. Meanwhile, citation overlap analysis provided deeper insights into the mean author cosine similarities achieved based on the author grouping strategy used.

The central focus of this study was the development and evaluation of a GNN model tailored specifically for link prediction tasks within the academic knowledge graph. The model architecture leveraged advanced techniques such as `SAGEConv` layers and heterogeneous transformations to capture local graph structures effectively. By aggregating information from a node's local neighborhood, the model could discern complex patterns and dependencies crucial for accurate link prediction.

Training procedures were optimized using mini-batch loading and edge-level splits to accommodate the scale and complexity of the knowledge graph, ensuring efficient processing and model scalability. The model's performance was evaluated across multiple metrics, including AUC, precision, recall, and F1-score. These metrics collectively demonstrated the model's high accuracy and reliability in predicting

links between authors and papers, highlighting its potential to uncover latent relationships.

The GNN model developed in this thesis represents a minimal yet sufficient framework that can be easily expanded to incorporate additional features and complexities. Future work could explore the integration of more detailed paper and author attributes to further enhance the model's predictive power. Additionally, the flexibility of the GNN architecture allows for the consideration of alternative layers and transformations, such as Graph Attention Networks (GAT) or Graph Convolutional Networks (GCN), to better capture the nuanced relationships within the knowledge graph.

The inclusion of more granular citation contexts and author collaboration networks could also improve the model's ability to predict emerging research trends and influential works. Furthermore, the development of more sophisticated training techniques, such as self-supervised learning and transfer learning, could enhance the model's scalability and applicability to larger, more diverse datasets.

Overall, the flexibility and extensibility of the GNN framework presented in this study pave the way for future advancements in the analysis of academic knowledge graphs. By continually refining and expanding upon this foundational model, researchers can unlock new opportunities for understanding and leveraging the complex web of academic relationships that drive scientific progress.

In conclusion, this thesis underscores the transformative potential of graph neural networks in modeling and analyzing complex relational data within academic knowledge graphs. By bridging the gap between data-driven insights and actionable knowledge, GNNs offer a powerful framework for advancing research in diverse fields, from recommender systems and information retrieval to the study of evolving academic landscapes. The culmination of this thesis serves as a testament to the capabilities of graph neural networks in unlocking hidden patterns and relationships within vast, interconnected datasets.

Appendix A

Figures

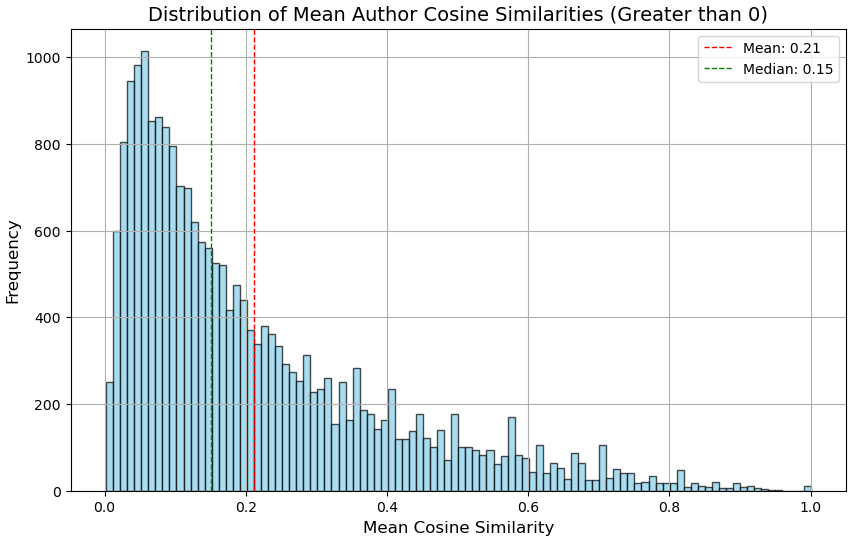

FIGURE A.1: Distribution of Mean Author Cosine Similarities for Full-Name Matching

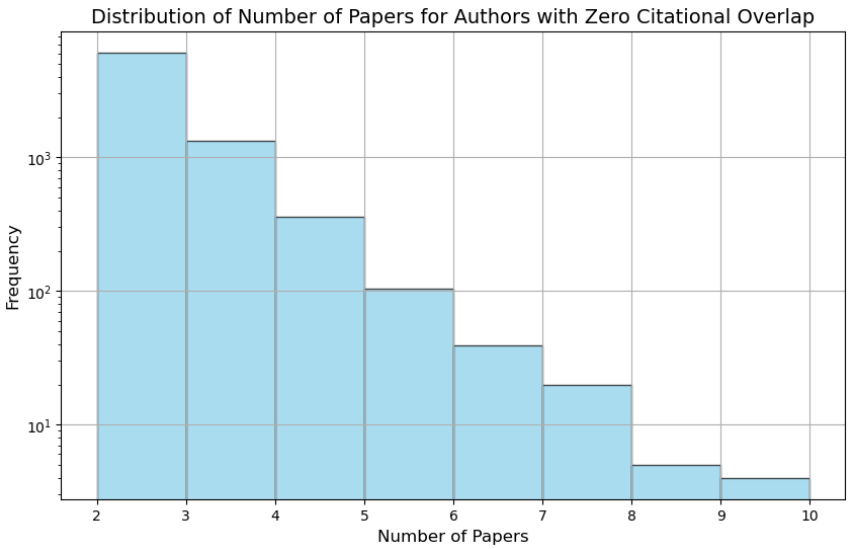

FIGURE A.2: Distribution of Number of Papers for Authors with Zero Citational Overlap for Full-Name Matching

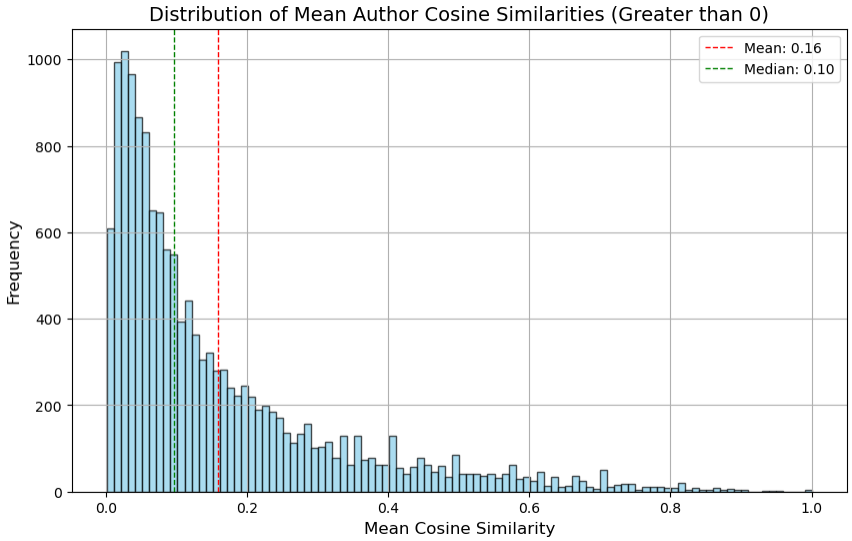

FIGURE A.3: Distribution of Mean Author Cosine Similarities for Surname Matching

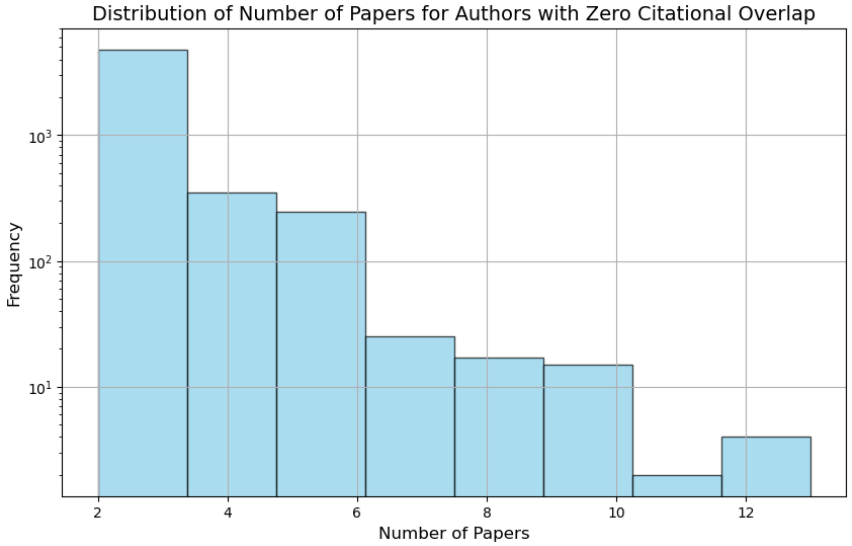

FIGURE A.4: Distribution of Number of Papers for Authors with Zero Citational Overlap for Surname Matching

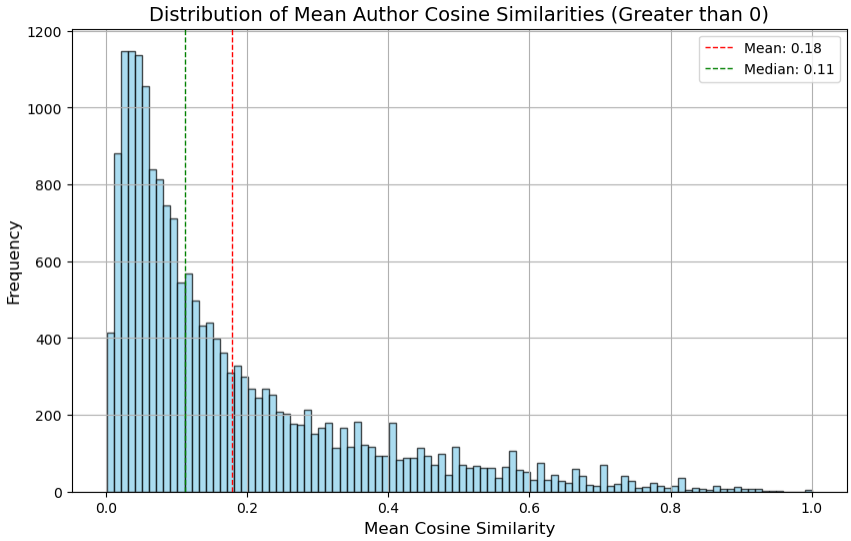

FIGURE A.5: Distribution of Mean Author Cosine Similarities for Normalized Indel Distance Matching

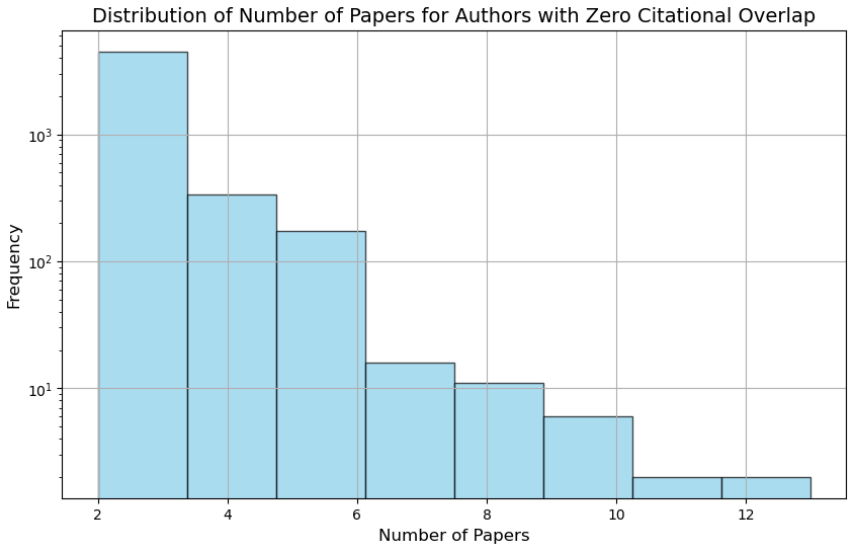

FIGURE A.6: Distribution of Number of Papers for Authors with Zero Citational Overlap for Normalized Indel Distance Matching

Bibliography

Adamic, Lada A and Eytan Adar (2003). "Friends and neighbors on the web". In: *Social Networks* 25.3, pp. 211–230. URL: https://doi.org/10.1016/S0378-8733(03)00009-1.

Angles, Renzo et al. (2017). "Foundations of Modern Query Languages for Graph Databases". In: *ACM Computing Surveys (CSUR)* 50.5, pp. 1–40. URL: https://doi.org/10.1145/3104031.

Bachmann, Maximilian (2024). *RapidFuzz: Fuzzy String Matching in Python*. Version 3.9.3. URL: https://github.com/maxbachmann/rapidfuzz.

Barabási, Albert-László and Réka Albert (1999). "Emergence of Scaling in Random Networks". In: *Science* 286.5439, pp. 509–512. URL: https://doi.org/10.1126/science.286.5439.509.

Bhattacharya, Indrajit and Lise Getoor (2007). "Collective entity resolution in relational data". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1, 5–es. URL: https://doi.org/10.1145/1217299.1217304.

Bishop, Christopher M (2006). *Pattern Recognition and Machine Learning*. Springer. DOI: 10.5555/1162264.

Blondel, Vincent D et al. (2008). "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10, P10008. URL: https://doi.org/10.1088/1742-5468/2008/10/P10008.

Bottou, Léon (2010). "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, pp. 177–186. URL: https://doi.org/10.1007/978-3-7908-2604-3_16.

Chawla, Nitesh V et al. (2002). "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16, pp. 321–357. URL: https://doi.org/10.1613/jair.953.

Christen, Peter (2011). "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication". In: *IEEE Transactions on Knowledge and Data Engineering* 24.9, pp. 1537–1555. URL: https://doi.org/10.1109/TKDE.2011.127.

Cohen, William, Pradeep Ravikumar, and Stephen Fienberg (2003). "A comparison of string metrics for matching names and records". In: *Kdd Workshop on Data Cleaning and Object Consolidation*. Vol. 3, pp. 73–78. URL: https://www.cs.cmu.edu/~wcohen/postscript/kdd-2003-match-ws.pdf.

Dean, Jeffrey and Sanjay Ghemawat (2008). "MapReduce: simplified data processing on large clusters". In: *Communications of the ACM* 51.1, pp. 107–113. URL: https://doi.org/10.1145/1327452.1327492.

Devlin, Jacob et al. (2018). "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805*. URL: https://doi.org/10.48550/arXiv.1810.04805.

Dong, Jianwei et al. (2022). "Process knowledge graph modeling techniques and application methods for ship heterogeneous models". In: *Scientific Reports* 12.1, p. 2911. URL: https://doi.org/10.1038/s41598-022-06940-y.

Dong, Xin Luna, Alon Halevy, and Cong Yu (2009). "Data integration with uncertainty". In: *The VLDB Journal* 18, pp. 469–500. URL: https://doi.org/10.1007/s00778-008-0119-9.

Ehrlinger, Lisa and Wolfram Wöß (2016). "Towards a Definition of Knowledge Graphs". In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4, p. 2. URL: https://ceur-ws.org/Vol-1695/paper4.pdf.

Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8, pp. 861–874. URL: https://doi.org/10.1016/j.patrec.2005.10.010.

Fey, Matthias and Jan E. Lenssen (2022a). *PyTorch Geometric Documentation*. URL: https://pytorch-geometric.readthedocs.io/en/2.5.0/generated/torch_geometric.data.HeteroData.html.

— (2022b). *PyTorch Geometric Documentation*. URL: https://pytorch-geometric.readthedocs.io/en/2.5.0/generated/torch_geometric.transforms.RandomLinkSplit.html.

— (2022c). *PyTorch Geometric Documentation*. URL: https://pytorch-geometric.readthedocs.io/en/2.5.2/_modules/torch_geometric/loader/link_neighbor_loader.html.

— (2022d). *PyTorch Geometric Documentation*. URL: https://pytorch-geometric.readthedocs.io/en/2.5.0/generated/torch_geometric.nn.conv.SAGEConv.html.

Fey, Matthias and Jan Eric Lenssen (2019). "Fast Graph Representation Learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428*. URL: https://doi.org/10.48550/arXiv.1903.02428.

Gilmer, Justin et al. (2017). "Neural Message Passing for Quantum Chemistry". In: *International Conference on Machine Learning*. PMLR, pp. 1263–1272. URL: https://doi.org/10.48550/arXiv.1704.01212.

Girvan, Michelle and Mark EJ Newman (2002). "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12, pp. 7821–7826. URL: https://doi.org/10.1073/pnas.122653799.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Grover, Aditya and Jure Leskovec (2016). "node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864. URL: https://doi.org/10.1145/2939672.2939754.

Hamilton, Will, Zhitao Ying, and Jure Leskovec (2017). "Inductive Representation Learning on Large Graphs". In: *Advances in Neural Information Processing Systems* 30. URL: https://doi.org/10.48550/arXiv.1706.02216.

Hirsch, Jorge E (2005). "An index to quantify an individual's scientific research output". In: *Proceedings of the National Academy of Sciences* 102.46, pp. 16569–16572. URL: https://doi.org/10.1073/pnas.0507655102.

Hogan, Aidan et al. (2021). "Knowledge Graphs". In: *ACM Computing Surveys (Csur)* 54.4, pp. 1–37. URL: https://doi.org/10.1145/3447772.

Jaccard, Paul (1901). "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines". In: *Bull Soc Vaudoise Sci Nat* 37, pp. 241–272. URL: http://dx.doi.org/10.5169/seals-266440.

Ji, Shaoxiong et al. (2021). "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2, pp. 494–514. URL: https://doi.org/10.1109/TNNLS.2021.3070843.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980*. URL: https://doi.org/10.48550/arXiv.1412.6980.

Kipf, Thomas N and Max Welling (2016). "Semi-Supervised Classification with Graph Convolutional Networks". In: *arXiv preprint arXiv:1609.02907*. URL: https://doi.org/10.48550/arXiv.1609.02907.

Li, Yuliang et al. (2020). "Deep Entity Matching with Pre-Trained Language Models". In: *arXiv preprint arXiv:2004.00584*. URL: https://doi.org/10.14778/3421424.3421431.

Liben-Nowell, David and Jon Kleinberg (2003). "The link prediction problem for social networks". In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 556–559. URL: https://doi.org/10.1145/956863.956972.

Mikolov, Tomas et al. (2013). "Distributed Representations of Words and Phrases and Their Compositionality". In: *Advances in Neural Information Processing Systems* 26. URL: https://doi.org/10.48550/arXiv.1310.4546.

Mudgal, Sidharth et al. (2018). "Deep Learning for Entity Matching: A Design Space Exploration". In: *Proceedings of the 2018 International Conference on Management of Data*, pp. 19–34. URL: https://doi.org/10.1145/3183713.3196926.

Murphy, Kevin P (2012). *Machine Learning: A Probabilistic Perspective*. MIT press. DOI: 10.5555/2380985.

OpenAlex (2024). *OpenAlex Dataset*. URL: https://openalex.org/.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems* 32. URL: https://doi.org/10.48550/arXiv.1912.01703.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). "Deepwalk: online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710. URL: https://doi.org/10.1145/2623330.2623732.

Powers, David MW (2020). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061*. URL: https://doi.org/10.48550/arXiv.2010.16061.

Prince, Simon J.D. (2023). *Understanding Deep Learning*. The MIT Press. URL: http://udlbook.com.

Provost, Foster J, Tom Fawcett, Ron Kohavi, et al. (1998). "The Case Against Accuracy Estimation for Comparing Induction Algorithms". In: *ICML*. Vol. 98, pp. 445–453. DOI: /10.5555/645527.657469.

Radicchi, Filippo, Santo Fortunato, and Alessandro Vespignani (2011). "Citation Networks". In: *Models of Science Dynamics: Encounters Between Complexity Theory and Information Sciences*, pp. 233–257. URL: https://doi.org/10.1007/978-3-642-23068-4_7.

Robbins, Herbert and Sutton Monro (1951). "A Stochastic Approximation Method". In: *The annals of Mathematical Statistics*, pp. 400–407. URL: https://doi.org/10.1109/TSMC.1971.4308316.

Robinson, Ian, Jim Webber, and Emil Eifrem (2015). *Graph Databases: New Opportunities for Connected Data*. "O'Reilly Media, Inc." DOI: 10.5555/2846367.

Schulz, Christian et al. (2014). "Exploiting citation networks for large-scale author name disambiguation". In: *EPJ Data Science* 3, pp. 1–14. URL: https://doi.org/10.48550/arXiv.1401.6157.

Society, American Physical (2023). "American Physical Society Dataset". In: *APS Journals*. URL: https://journals.aps.org/datasets.

Tang, Jie et al. (2008). "Arnetminer: extraction and mining of academic social networks". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 990–998. URL: https://doi.org/10.1145/1401890.1402008.

Wu, Zonghan et al. (2020). "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1, pp. 4–24. URL: https://doi.org/10.1109/TNNLS.2020.2978386.

Xu, Jian et al. (2020). "Building a PubMed knowledge graph". In: *Scientific Data* 7.1, p. 205. URL: https://doi.org/10.1038/s41597-020-0543-2.

Zhang, Muhan and Yixin Chen (2018). "Link Prediction Based on Graph Neural Networks". In: *Advances in Neural Information Processing Systems* 31. URL: https://doi.org/10.48550/arXiv.1802.09691.

Zhao, Zhongying et al. (2018). "A comparative study on community detection methods in complex networks". In: *Journal of Intelligent & Fuzzy Systems* 35.1, pp. 1077–1086. URL: https://doi.org/10.3233/JIFS-17682.

Zhou, Jie et al. (2020). "Graph neural networks: A review of methods and applications". In: *AI Open* 1, pp. 57–81. URL: https://doi.org/10.1016/j.aiopen.2021.01.001.