Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Eric Wonhee Lee Date

Automating Biomedical Abstract Screening using Network Embedding

By

Eric Wonhee Lee
Doctor of Philosophy

Computer Science and Informatics

Joyce C. Ho, Ph.D.
Advisor

Byron C. Wallace, Ph.D.
Committee Member

Carl Yang, Ph.D.
Committee Member

Li Xiong, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Automating Biomedical Abstract Screening using Network Embedding

By

Eric Wonhee Lee
B.S., University of Illinois at Urbana-Champaign, IL, 2011
M.S., Yonsei University, South Korea, 2017

Advisor: Joyce C. Ho, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Abstract

Automating Biomedical Abstract Screening using Network Embedding
By Eric Wonhee Lee

Systematic review (SR) is an essential process to identify, evaluate, and summarize the findings of all relevant individual studies concerning health-related questions. However, conducting a SR is labor-intensive, as identifying relevant studies is a daunting process that entails multiple researchers screening thousands of articles for relevance. Automating SR, especially abstract screening, using machine learning models has been proposed to identify relevant articles but primarily focuses on the text and ignores additional features like citation information. Recent work demonstrated that citation embeddings can outperform the text itself, suggesting that better network representation may expedite SRs. Yet, how to utilize the rich information in heterogeneous information networks (HIN) for network embeddings is understudied. Also, the lack of a unified source that includes the metadata of biomedical literature makes the research more challenging. To deal with this problem, we propose four works. First, we propose a model that exploits three representations, documents, topics, and citation networks to show the effectiveness of the additional features. Second, we introduce the PubMed Graph Benchmark, one of the largest HIN to date, which aggregates the rich metadata into a unified source that includes abstracts, authors, citations, MeSH terms, etc. Third, we propose a HIN embedding model that uses a community-based multi-view graph convolutional network for learning better representations using the PubMed Graph Benchmark. Lastly, we propose a hyperbolic representation learning model for graphs with mixed hierarchical (MeSH hierarchies) and non-hierarchical (citations) structures.

Automating Biomedical Abstract Screening using Network Embedding

By

Eric Wonhee Lee
B.S., University of Illinois at Urbana-Champaign, IL, 2011
M.S., Yonsei University, South Korea, 2017

Advisor: Joyce C. Ho, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2023

Acknowledgments

I would like to thank my esteemed supervisor - Dr. Ho for her invaluable supervision, support, and tutelage during the course of my Ph.D. degree. Additionally, I would like to express gratitude to Dr. Wallace and Dr. Yang for their treasured support which was really influential in shaping my research methods and critiquing my results. I would also like to thank Dr. Xiong for her impressive discussions of my work and inspiring suggestions for this dissertation.

Thanks my friends, lab mates, colleagues, and research team for a cherished time spent together in the lab, and in social settings. My appreciation also goes out to my family and friends for their encouragement and support throughout my studies.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Systematic reviews (SRs) are essential knowledge translation tools focused on bridging the research-to-practice gap across a wide range of domains. In health research, SRs aim to identify, evaluate, and summarize the findings of all individual studies (which typically describe clinical trial results) relevant to a clinical question, thereby making the available evidence more accessible. For instance, a SR can be used to synthesize findings from randomized intervention studies to robustly determine which interventions are best supported for a particular condition. Thus SRs (and meta-analyses) provide high-quality evidence that can inform healthcare decision-making, support clinical guidelines, and guide health policies [17, 37, 38].

Unfortunately, conducting SRs is a time-consuming and complex task [43]. Established methodologies for performing a SR [20, 78, 81] require a comprehensive search to identify all the relevant studies for inclusion. Indeed, comprehensiveness (so as to avoid bias via 'cherry-picking' of evidence) is a key property of rigorous evidence synthesis. Yet the broad searches necessary to achieve this yield imprecise search results including searches that often yield only $\sim$1% relevant results. Domain experts

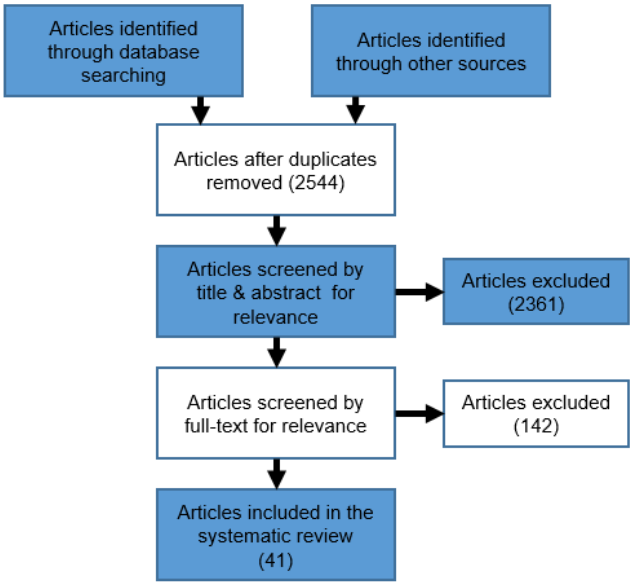


Figure 1.1: A simplified illustration of the SR screening process using "ACEInhibitors" from Cohen [26] dataset.

must wade through these mostly irrelevant articles to identify those that meet the *inclusion criteria*; thus, producing a single review can require thousands of person-hours [2]. Figure 1.1 provides an example of the laborious SR screening process for ACE inhibitors. The step for identifying the initial list which is 2544 from the figure is called the *Identification* step of the SR screening process. The step for determining relevant articles from the title and the abstract is the *Abstract Screening* step, and from the full-text is the *Eligibility* step. Only 1.61% of the articles were selected from the *Eligibility* step, and only 7.19% were included (i.e., analyzed and evaluated) from the *Abstract Screening* step in the actual review itself. The estimate for the average time to conduct a SR is 67 weeks from registration to publication [13]. Clearly, this process is unsustainable nor scalable, especially given the exponential growth of biomedical literature [8].

Given the importance of SRs for realizing evidence-based practice and the labor that conducting these entails, there is a clear need to expedite tasks necessary for evidence synthesis while maintaining rigor and comprehensiveness. In particular, semi-

automation can help speed up the screening process, an extremely time-consuming endeavor due to a large number of articles [80]. The standard methodology for semi-automating the abstract screening step of SRs entails training a custom classification model for each new review. Unfortunately, many of the previous approaches assume that they have small labeled batches from the reviewers, and train their model on those batches to predict the rest [26, 57]. Moreover, the existing methods focus primarily only on the text itself using representations like bag-of-words or word embeddings [9, 25, 26, 57, 61, 77, 117]. However, there is rich information (i.e., citation relationships between the articles, or MeSH Terms) that can be used to learn more accurate models. Khabsa *et al.* [57] used co-citation relation between articles with brown clustering on the bi-grams to tackle the data sparsity problem but still the usage of citations remains largely underexplored.

Citation networks can be represented as a graph structure that includes articles (nodes) and references (edges). This representation is used across many application domains including social networks, the World Wide Web, and knowledge graphs. As real-world networks can be huge and complex, it is difficult to analyze the graph, thus learning meaningful low-dimensional vectors of the nodes and edges or network embeddings have been proposed while preserving the features of the network [70]. Recently, there has been an emergence of deep learning-based models such as graph neural networks (GNN) to learn the network embeddings [36, 71, 100]. One popular method is Graph Convolutional Network (GCN) [59] which can efficiently learn the structural dependencies through convolutional operations on the graph. However, GCN is designed for a homogeneous network that contains a single node type and edge type as a citation network (only paper nodes and paper-paper edges in the citation network). The bibliographic network of biomedical literature is a heterogeneous information network (HIN) and contains multiple objects (nodes) and link types (edges) including citations, author information, venue information, and Medical

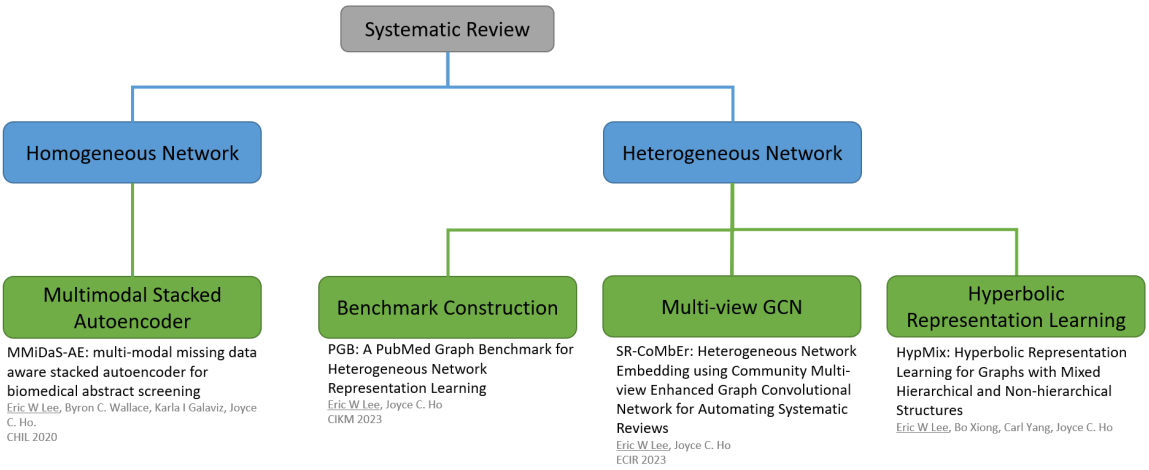


Figure 1.2: Dissertation Contributions.

Subject Headings (MeSH) terms that are used for indexing articles.

Since many real-world networks are HIN with multiple objects and link types, several variations of GNN and GCN models have been proposed for HIN embeddings. However, existing models have focused on preserving the meta-path structure (i.e., the path with various object types and edge types that capture the semantics of the network) by transforming the HIN into several homogeneous networks to learn the representations [31, 102, 134]. Unfortunately, the defined meta-path impacts the embedding quality. ie-HGCN [132] was introduced to automatically evaluate all possible meta-paths and project the representations of different types of neighbor objects into a common semantic space. However, ie-HGCN is susceptible to noise in the graph and ignores the community structure.

The goal of this dissertation is to reduce the reviewers' workload by excluding the maximum number of irrelevant documents in the abstract screening step by leveraging the rich information in the HIN bibliographic network. Our general strategy builds upon a body of work on semi-automating abstract screening for evidence syntheses via machine learning [26, 89, 117, 119], but focuses on layering in citation information and other metadata to improve the automation process.

1.2 Research Contributions

Given the limitations of existing semi-automated biomedical abstract screening methods of SR to incorporate citation networks, we propose new models to (1) appropriately leverage the co-citation information with the textual data to learn better representations, (2) incorporate the different edge and node types into a unified embedding that captures the complex structure, and (3) utilizes the hierarchical structure of the MeSH terms associated with each article to improve the article embedding. We also introduce a new benchmark dataset for evaluating HIN embeddings that layer in the rich metadata in PubMed. Our contributions (illustrated in Figure 1.2) are summarized as follows.

1.2.1 MMiDaS-AE

We propose MMiDaS-AE which adopts multi-modal stacked autoencoder [15] to encode three input representations, document, topic, and citation. The multi-modal stacked autoencoder learns a shared representation of different inputs that have different information. The document representation uses the text from the title and the abstract, the topic representation uses MeSH terms, and citations use the co-citation relations of the articles. As the citation information is not complete, we also propose a missing data imputation technique to handle any missing information.

1.2.2 PGB

We present PGB, a new benchmark dataset of over 30 million PubMed articles for evaluating HIN embeddings for biomedical literature. It leverages the citations and author disambiguation capabilities of Semantic Scholar while also layering in the rich metadata that is offered in PubMed including the MeSH Terms, Chemical List, and Publication Type. PGB also layers in the MeSH hierarchical structure for all the

terms associated with the articles, which previous benchmarks do not support.

1.2.3 SR-CoMbEr

We propose SR-CoMbEr, a HIN embedding model, which constructs multiple local GCNs where each of them is centered around a community. To learn from the different object and link types, each community adopts a multi-view approach where a view-specific representation is learned to capture the complex structure information for each relation type. Moreover, we pose the multiple community GCN aggregation problem as a multi-modal problem to yield a robust final embedding that reflects the different community representations.

1.2.4 Hyperbolic Representation Learning

We propose HypMix, a new hyperbolic representation learning model to capture graphs with mixed hierarchical (MeSH terms) and non-hierarchical (citations) structures of the articles. MeSH terms help identify the article contents and codify abstract concepts that can be divided into subcategories that are arranged from general to specific such as $Head \rightarrow Face \rightarrow Eye$. Each MeSH category can contain up to 13 hierarchical levels. To capture the MeSH hierarchy, we adopt Poincaré embedding space [86] and introduce regularizations and hyperbolic entail cones [5] to represent not only the hierarchical structure but also the non-hierarchical relationships between MeSH terms and articles.

1.3 Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces the basic background of SR and related works. Chapter 3 proposes MMiDaS-AE, a multi-modal stacked autoencoder model that uses three different sources of information.

Chapter 4 introduces PGB, a PubMed graph benchmark which is a new benchmark dataset that includes the rich metadata of biomedical literature. Chapter 5 describes SR-CoMbEr, a community-based multi-view GCN to capture the structural heterogeneity that is useful for downstream tasks. Chapter 6 proposes HypMix, a hyperbolic representation learning model for mixed hierarchical and non-hierarchical structures which uses the MeSH hierarchies and citations. Finally, Chapter 7 concludes the dissertation and discusses the future direction.

Chapter 2

Background and Related Work

2.1 Systematic Review

Methods for semi-automating the abstract screening step of SRs have been widely studied; see [89] for a survey of this work. The typical approach is to adopt a supervised learning model – equivalent to training a custom classification model for each new review. The classification models used to discriminate between relevant and irrelevant articles for a given topic include support vector machines (SVMs) [46, 91, 117, 119], generalized linear models [48], Voting Perceptron [26], Random Forest [57], Complement Naive Bayes [75], Decision Tree [9], and k-NN [1]. Note that models can be used either to make 'hard' include/exclude decisions, or can be used to rank articles in order of likely relevance.

Because supervision is expensive for this task, and a new model must be trained for each new review, a common strategy explored is active learning [27, 61, 77, 116, 119] in which the learner starts with a small subset of manually labeled records, which are used to train the initial classifier. After each learning (or annotation) cycle, the newly trained model classifies the remaining unlabelled citations and presents a sample of these records to the reviewer for annotation. This iterative approach may be used to

train a model that is used to classify all remaining (unscreened) articles, or can simply be used to prioritize identification of relevant abstracts so that the review team can begin data extraction from these [89].

Most of the semi-automation SR approaches use bag-of-words and their combinations [9, 25, 26, 57, 61, 77, 117]. For example, Cohen *et al.* [25] proposed to use uni-grams and bi-grams to treat each of them as a single word, Bannach-Brown *et al.* [7] used tri-gram and GENIA tagger [110] prior to extracting uni-grams, and Khabsa *et al.* [57] used brown clustering on the bi-grams to tackle the data sparsity problem. Some more recent efforts have proposed to learn a paragraph vector using neural models [46, 69].

2.2 Network Embedding

The goal of network embedding (or network representation learning) is to learn low-dimensional vectors that are projected into an Euclidean space while preserving the network structure and the property. Within the past few years, many network embedding methods have been proposed. For example, node2vec [40] and DeepWalk [95] used random walk-based method, SDNE [120] used deep neural network-based method, and M-NMF [121] used matrix factorization based method. However, all these methods are introduced for the homogeneous network.

A heterogeneous information network (HIN) contains multiple types of objects and links. Formally, such a network is defined as follows.

Definition 1. HIN. A HIN is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi, \psi)$, where $\mathcal{V}$ is the set of objects, $\mathcal{E}$ is the set of links, ϕ is the object type mapping function, and ψ is the link mapping function. ϕ is defined as $\phi : \mathcal{V} \rightarrow \mathcal{A}$, and ψ is defined as $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denotes predefined object and link types respectively where $|\mathcal{A}| + |\mathcal{R}| > 2$.

A homogeneous network contains a single object and relation type such as a social

network with User (U) as an object type and a single type of link U – U. On the other hand, HIN contains multiple types of objects such as a bibliographic network which has four types of objects (i.e., Author (A), Paper (P), Venue (V), and MeSH terms (M) and three link types, A – P, P – V, and P – M.

2.2.1 Graph Neural Networks

Graph Neural Networks (GNNs) are a conventional model that has been widely studied in various tasks using graphs in recent years. GNNs extend the deep neural network to transform the complicated graph-structured data into a meaningful representation while preserving the graph structure. Graph Convolutional Network (GCNs) is an extension of GNN and has been proven to be efficient in achieving a good performance in a variety of graph datasets. GCNs can be categorized into two categories: 1) spectral [14, 29, 47, 59, 68, 128] and 2) non-spectral [21, 44, 79, 112]. For the spectral approach, Bruna *et al.* [14] proposed to do convolution in the spectral domain by using a Fourier basis. Kipf *et al.* [59] proposed a convolutional architecture via a localized first-order approximation of spectral graph convolutions. Michaël *et al.* [29] proposed to use K-order Chebyshev polynomials to approximate smooth filters in the spectral convolutions. On the other hand, non-spectral approaches define convolution operations directly on the graph operating on spatially close neighbors. For example, Hamilton *et al.* [44] proposed GraphSAGE which uses local aggregation functions from sampled local neighbors for the target node. Also, attention mechanisms are used in GNNs. For example, Veličković *et al.* [112] proposed to use self-attention to enable specifying different weights to different nodes in a neighborhood. However, these models cannot deal with multiple types of nodes and edges, in other words, can only be used for homogeneous networks.

2.2.2 Graph Convolutional Networks

GCNs [59] can be formally defined as follows. Suppose H^k is the feature representation of the k-th layer in GCN, the propagation becomes

$$H^k = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{k-1}W^k) \tag{2.1}$$

where $\tilde{A} = A + I \in \boldsymbol{R}^{N\times N}$ is the adjacency matrix A with a self connection. $\tilde{D}$ is the degree matrix of $\tilde{A}$ which is formally defined as $\tilde{D}_{ii} = \sum_i \tilde{A}_{ij}$. And W^k is a trainable weight matrix. As shown in Equation (2.1), the convolution operation is determined by the given graph structure and GCN only learns the node-wise linear transform $H^{k-1}W^k$. Thus, the convolution layer can be interpreted as the composition of a fixed convolution followed by an activation function σ on the graph after the node-wise linear transformation.

2.2.3 Heterogeneous Information Network Embedding

For HIN embedding, many works focus on preserving the meta-path structure which contains the semantic information of the graph. ESim [102] uses multiple user-defined meta-path to learn representations in the user-preferred embedding space, and metapath2vec [31] is a skip-gram model that uses meta-path based random walk. HIN2Vec [32] learns the representation of nodes and meta-paths to capture the semantics in the network by carrying out multiple prediction training tasks, and HERec [103] captures the semantics in the network by using the type constraint strategy to filter the node sequence. Some works extend GNNs for modeling HIN. For example, HAN [122] proposed to transform the given HIN into a homogeneous network based on the meta-path and use GNN based on the hierarchical attention.

HIN embedding models which encode hierarchical structure are also studied. Yang *et al.* [130] proposed TAXOGAN which is a model that co-embeds nodes with a

non-hierarchical structure and taxonomies with a hierarchical structure. Bai *et al.* proposed ConE for a knowledge graph that has heterogeneous hierarchies using a relation-specific transformation. However, limited work encodes both hierarchical and non-hierarchical structures for the heterogeneous graph representation learning task. Most of the works target learning the representations of only hierarchical structures for topic detection [125, 137], taxonomy completion [4, 54], and recommendation systems [127]. To handle the graph with mixed hierarchical and non-hierarchical structures, Gu *et al.* [41] proposed a model that uses the mixed-curvature representations of the product of hyperbolic and spherical space.

2.3 Hierarchical Structure Embeddings

Hierarchical structures are common in the real-world, for example, social networks, sentences in natural language, and evolutionary relationships in phylogenetics [94]. One important characteristic of hierarchical structures is that the number of leaf nodes increases exponentially as the number of hierarchical levels increases. As such, most graph representation learning approaches suffer from distortion issues when embedding graphs with hierarchical structures. Instead, hyperbolic space has been proposed as an alternative to learning the latent hierarchical structures in the context of embedding models. Hyperbolic representation learning is widely studied in the area of knowledge graph [5, 19, 106], lexical entailment [42, 86, 98], and recommender systems [22, 114]. The goal of the model is to learn the representation of the hierarchical structure which can be represented as a tree-like structure.

Hyperbolic representation learning models use the hyperbolic space to learn the representation of the underlying hierarchical structures [131]. One important characteristic of the tree-like structure is that the number of leaf nodes increases exponentially as the number of levels (or tree depth) increases. This is suboptimal for

Euclidean space as the volume of the ball increases polynomially according to the radius, r, such that $V_d^{\mathbb{E}}(r) = \Theta(r^d)$. As a result, Euclidean space becomes too narrow to accommodate the exponential leaf node growth that arises with increasing tree depth.

On the other hand, the hyperbolic space can handle this growth. The equation of the hyperbolic (or Poincaré) ball volume is:

$$V_2^{\mathbb{H}}(r) = \Theta(e^r) \tag{2.2}$$

where r is the radius. Another benefit of hyperbolic space is when computing the distance between two nodes. The distance between two leaf nodes can be small when using the Euclidean distance, however, in the hyperbolic distance we can measure the tree distance which helps to capture the property of the tree-like graph. Thus, hyperbolic space allows learning representations of symbolic data by capturing hierarchy and similarity.

2.3.1 Poincaré Embedding

Poincaré embedding [86] is a model for learning representations of hierarchical structure. It uses hyperbolic space or an n-dimensional Poincaré ball and shows advantages over Euclidean embedding on hierarchical data. Poincaré Embedding is based on the approach of the Poincaré ball model which is well-suited for gradient-based optimization.

Let $B = x \in \mathbb{R}^d | ||x|| < 1$ be the open d-dimensional unit ball where $|| \cdot ||$ denotes the Euclidean norm. The Poincaré ball model of hyperbolic space uses Riemannian manifold (B^d, g_x). The open unit ball equipped with the Riemannian metric tensor is

$$g^B = (\frac{2}{1 - ||x||^2})^2 g^E \tag{2.3}$$

where $x \in B^d$ and g^E denotes the Euclidean metric. The distance between two nodes $u, v \in B^d$ in Poincaré ball model is

$$d(u,v) = arccosh(1 + \frac{2||u-v||^2}{(1-||u||^2)(1-||v||^2)}) \tag{2.4}$$

The boundary of the Poincaré ball is denoted by ∂B. The geodesics in B^d are then circles that are orthogonal to ∂B.

2.3.2 Hyperbolic Entailment Cones

Hyperbolic entailment cones [33] is a model that embeds nodes into hyperbolic cones which is a generalized idea of order embedding [113]. The hyperbolic entailment cones model views the hierarchical relations as partial orders and is defined by a family of nested geodesical cones. In short, it defines a region that is the hierarchical relation that can fit in the hyperbolic space. Using this idea, it addresses the limitation of Poincaré embedding model which is that most points collapse on the border of the Poincaré ball.

Let C_x denote the cone at apex x. The goal is to model partial order by containment relationship between cones. The cones satisfy transitivity:

$$\forall x, y \in B^d 0 : y \in C_x \Rightarrow C_y \subseteq C_x \tag{2.5}$$

For $x, y \in B^d$, the angle of y at x to be the angle between the half-lines $\overrightarrow{ox}$ and $\overrightarrow{xy}$ and denote it as $\angle_x y$. This can be expressed as:

$$\angle_x y = \cos^{-1}(\frac{< x, y > (1+x^2) - ||x||^2(1+||y||^2)}{||x||||x-y||\sqrt{1+||x||^2||y||^2 - 2 < x, y >}}) \tag{2.6}$$

To satisfy the transitivity of nested angular cones and symmetric conditions [33], the

following expression of Poincaré entailment cone at apex $x \in B^d$ can be used.

$$C^x = y \in B^d | \angle_x y \leq \sin^{-1}(K \frac{1 - ||x||^2}{||x||}) \tag{2.7}$$

where $K \in R$ is a hyperparameter.

2.4 Systematic Review Datasets

For ease of comparison with previous works, we evaluate our model on the publicly available dataset provided by Cohen *et al.* [26]. The dataset includes 15 SRs (or topics) concerning different drug efficacies.[1] The 15 SRs were performed by members of evidence-based practice centers (EPCs). Each SR contains a PubMed identifier (PMID), and abstract triage status. The PMID allows us to identify which article was included in the SR process. Abstract triage status indicates whether the article passed the title/abstract screening.

The next set of datasets is 3 sets provided by SWIFT-Review [49].[2] The dataset was generated by the National Toxicology Program (NTP) Office of Health Assessment and Translation (OHAT). The next 3 sets are provided by CLEF 2019 e-Health TAR Lab [55] (Task 2)[3] which focuses on retrieving relevant studies from during the abstract phase of conducting an SR. From the CLEF-TAR dataset, we randomly selected 3 sets which are the CD012661 topic from Prognosis, the CD008803 topic from DTA, and the CD005139 topic from Intervention. The title of the topic CLEF-Prognosis-CD012661 is "Development of type 2 diabetes mellitus in people with intermediate hyperglycemia" [97]. The title of the topic CLEF-DTA-CD008803 is "Optic nerve head and fibre layer imaging for diagnosing glaucoma"[76]. The title of the

[1] This dataset was later extended to include 24 SRs [27], however, only 15 SRs have been made publicly available.

[2] `https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-016-0263-z#Sec30`

[3] `https://github.com/CLEF-TAR/tar`

Table 2.1: Statistics of all datasets used. Abs refers to the number of abstract triages. % shows the percentage of the articles that are included after the abstract screening.

Dataset	Abs	Total	%
Cohen-ACEInhibitors	183	2544	7.19
Cohen-ADHD	84	851	9.87
Cohen-Antihistamines	92	310	29.67
Cohen-AtypicalAntipsychotics	363	1120	32.41
Cohen-BetaBlockers	302	2072	14.57
Cohen-CalciumChannelBlocker	279	1218	22.90
Cohen-Estrogens	80	368	21.74
Cohen-NSAIDs	88	393	22.39
Cohen-Opioids	48	1915	2.51
Cohen-OralHypoglycemics	139	503	27.63
Cohen-ProtonPumpInhibitors	238	1333	17.85
Cohen-SkeletalMuscleRelaxants	34	1643	2.56
Cohen-Statins	173	3465	4.99
Cohen-Triptans	218	671	32.48
Cohen-UrinaryIncontinence	78	327	23.85
SWIFT-Transgenerational	765	48638	1.57
SWIFT-PFOS-PFOA	95	6331	1.50
SWIFT-BPA	111	7700	1.44
CLEF-Prognosis-CD012661	192	3367	5.70
CLEF-DTA-CD008803	99	5220	1.89
CLEF-Intervention-CD005139	112	5392	2.07
Anemia	653	5653	11.55
COPD	196	1606	12.20
Clopidogrel	771	8291	9.30
Proton Beam	243	4751	5.11

topic CLEF-Intervention-CD005139 is "Anti-vascular endothelial growth factor for neovascular age-related macular degeneration" [105]. Last 4 datasets we use are: COPD [23], proton-beam [108], anemia [62], and clopidogrel [6].

Table 2.1 reports the distribution of articles in each topic. The first 15 SRs are the Cohen dataset, the next three are the SWIFT-Review dataset, the next three are the CLEF-TAR dataset, and the last four are from other sources. As shown in the table, for the abstract screening results, the number of articles screened ranged from 310 (Antihistamines) to 48,638 (Transgenerational) with anywhere from 1.44% (BPA) to

32.48% (Triptans) passing the abstract screening process. This demonstrates a large degree of imbalance.

2.5 Evaluation Metrics

Cohen *et al.* [26] introduced a new measure *work saved over sampling* (WSS). WSS measures the work saved over random sampling for a given level of recall. WSS is defined as

$$WSS = (TN + FN)/N - (1.0 - R) \tag{2.8}$$

where TN denotes true negatives, FN denotes false negatives, N is the total number of articles, and R is the recall. Cohen *et al.* used the special modification of the WSS called WSS@95% which means WSS for recall at 95%. Note that in some cases, the models may not achieve exactly 95% recall. Thus, to calculate WSS@95%, we compute WSS with the highest recall of no less than 95%. In addition to WSS@95%, some works reported an area under the receiver operating curve (AUC); we use this as an additional evaluation metric.

Chapter 3

Multi-modal Missing Data aware Stacked Autoencoder

Our general strategy builds upon a body of work on semi-automating abstract screening for evidence syntheses via machine learning [26, 89, 117, 119]. To address the limitations of existing semi-automation SR models, we introduce MMiDaS-AE [67], a **M**ulti-modal **Mi**ssing **D**ata **a**ware **S**tacked **A**uto**E**ncoder. We adopt the multi-modal stacked autoencoder [15] to encode a variety of information that includes 1) text from the document, 2) Medical Subject Headings (MeSH) terms, and 3) citation networks. In addition to the textual data in the documents, each article in PubMed (a repository of biomedical articles) is associated with MeSH terms, which codify abstract concepts and can be used to learn topic representations. MMiDaS-AE also uses co-citation relations between articles. The intuition is that an unknown article with co-citation relations to an article that passes the SR screening is more likely to be relevant.

However, it is crucial for the model to be robust to missing data representations, especially when learning a shared representation using three different sources of information. Thus, to mitigate the effects of missing data, we extend work for bimodal

speech classification [85] to design an imputation technique for multi-modal data in which we intentionally leave one or more representations out while learning to induce a shared representation in a latent space from which we can reconstruct all input modalities. Consequently, this multi-modal stacked autoencoder is robust to missing data. We also introduce a multi-label classification task to improve the prediction result by utilizing whether the article passed the abstract screening and whether it passed the full-text screening. Finally, we utilize a cross-topic learning strategy to utilize existing SRs to pre-train MMiDaS-AE, and then fine-tune the weights of the model to a specific SR topic.

3.1 Feature Representations

Feature extraction is a crucial component of the success of the classification process. Previous approaches use bag-of-words of titles, abstracts, and MeSH terms [9, 25, 26, 57, 61, 77, 117]. Khabsa *et al.* [57] used co-citation data as a feature to semi-automate the SR. However, unlike previous approaches that deal with each representation separately, we propose to learn a shared representation that encodes different article information. As a result, the model can be robust to missing data and a limited number of samples.

3.1.1 Document Representation

Natural language processing (NLP) systems typically transform input documents into fixed-dimensional vector representations that can subsequently be used as feature vectors by 'downstream' modules (e.g., logistic regression or a feed-forward neural network). Previous work for semi-automating screening for SRs predominantly represented documents via sparse bag-of-words (BOW) representations [7, 26, 57, 75, 88, 117]. More recent work in NLP has moved towards learning better representations

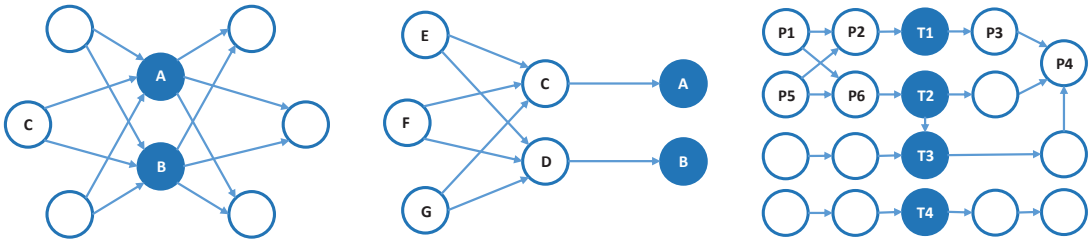
of texts, in particular by mapping high-dimensional and sparse BOW representations into dense, low-dimensional vectors. For example, doc2vec extends word2vec to learn distributed representations of documents (rather than words) [56, 64]. ELMO [96] and BERT [30] were proposed to learn the contextual representations.

For our task, we restrict our document to titles and abstracts due to potential copyright issues inherent to full-text articles. As a result, each article's input is relatively short (an average of 118 words after simple preprocessing). For short texts, averaging the embeddings of all words in the text can serve as the document representation [56]. Therefore, we adopt PMCVec [34], a pre-trained word2vec embedding, and learn the document representations of all articles by averaging embeddings in the title and the abstract of each document. PMCVec was trained on titles and abstracts from ∼27 million documents indexed in the PubMed database. We explored SciBert [10], a deeper representation, but this did not yield better predictive power as demonstrated in our empirical results.

3.1.2 Topic Representation

In the *Identification* step of SRs depicted in Figure 1.1, a combination of MeSH terms that represent the SR topic is used for database search to retrieve the initial articles list. Using these MeSH terms, we can compute the distance between the article and the MeSH terms. Thus, we learn a topic representation of an article by using the relationship between MeSH terms and the article. This can be done in two steps. First, we learn the representation of all MeSH terms of the topic. Second, we subtract the document representation we learned from the previous section from the MeSH term representation. Thus, this topic representation captures the relationship between the article and the MeSH terms used in the SR search. This has the added benefit of distinguishing articles that are in multiple SRs.

Because we are learning the relationships between documents and associated



(a) Co-citation relation (b) Two-step co-citation relation (c) Partial citation network

Figure 3.1: A simplified example of the co-citation relations and the partial citation network used to learn representations. Solid nodes denote target articles that we need to classify; empty nodes are the articles used to learn the representation of target articles.

MeSH terms of the topic by subtracting their representations, both representations should be learned from the same embedding space. One benefit of PMCVec [34] is that it learns representations of both single words and multi-words from PubMed abstracts as technical phrases in biomedical texts such as diseases or symptoms are multi-word phrases. Thus for MeSH terms, instead of using the composition of single words, multi-word MeSH terms also appear in the embedding space, and we can directly use them to compute the MeSH terms embedding.

3.1.3 Citation Network Representation

Most existing SR screening methods primarily rely on text features derived from titles and abstracts. This ignores the rich citation structure (e.g., the study is cited by other studies) available for each article. Figure 3.1(a) depicts a simple citation network (a network in which articles are nodes and citations are edges), and C and A implies A is cited by C (or C is a citation of A). From the citation network, co-citations (two articles cited together by the other articles) might be used to find related studies. For example, in 3.1(a), A and B are co-citations as there exists an article C that cites both A and B. This is motivated by the intuition that if one article is included, then co-cited articles are more likely to be included as well. Using features consisting of

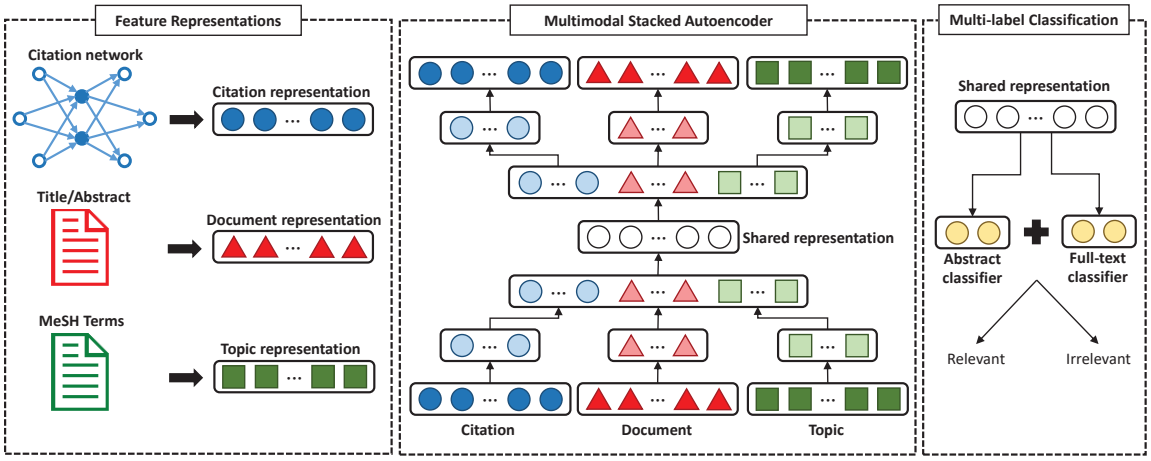


Figure 3.2: Framework overview of the MMiDaS-AE.

just the bag-of-words of uni-grams and co-citations, Khabsa *et al.* [57] showed that their machine learning model could achieve good recall. Yet, co-citation only captures one perspective of the article. There exist cases in which two articles do not have a co-cited article, but their citations have co-cited articles. Therefore, we propose to construct a citation network and learn a representation (low-dimensional projection) of each article.

However, constructing a complete citation network is infeasible. Instead, we use a partial citation network that contains co-citation information by limiting the network to contain only articles at most two citations away.

3.2 Model design

MMiDaS-AE adopts a multi-modal stacked autoencoder [15] which takes multiple input representations and learns a shared representation that encodes all of these modalities. This avoids the unwieldy number of parameters that are introduced with a simple concatenation of each input representation. Also, compressing the feature representations into a shared representation makes it easier to apply any matrix manipulation technique that can not be done in the input space because of

the difference in dimensions. However, the existing work was insufficient to deal with missing data representations. Thus we introduce a new learning strategy by using an augmented dataset. Finally, we propose a multi-label classification task to improve the prediction results. An illustration of our framework is shown in Figure 3.2.

3.2.1 Multi-modal Stacked Autoencoder

Autoencoders are unsupervised models that learn compressed representations of inputs. The objective for the autoencoder is to reconstruct inputs faithfully from this learned representation with minimal error [93]. Cadena *et al.* [15] proposed multi-modal stacked autoencoders for the task of robotics scene understanding to support different input modalities simultaneously (i.e., RGB image, scene depth, and semantic information). Each input representation was passed through an autoencoder. The three independent autoencoders were concatenated together using their respective hidden layers and then passed to another autoencoder, thus inducing a shared representation from which to reconstruct the original (concatenated) inputs. One may view this approach as a means of learning *disentangled* representations [51, 74] in which we have explicit low-dimensional encodings of the respective input modalities. We found empirically that the best performance was obtained when we unified the length of the independent hidden layer prior to concatenation. For example, if we have 256, 200 and 200 dimensions as an input for each representation, the best performance we get is when we use unified (e.g. 100) dimensions for the independent hidden layer.

3.2.2 Missing Data Imputation in Autoencoder

One advantage of multi-modal auto-encoders is their potential to combine the available modalities to impute representations in the case that one of these is missing [15, 85]. However, robustness to missing data is crucial when learning the shared rep-

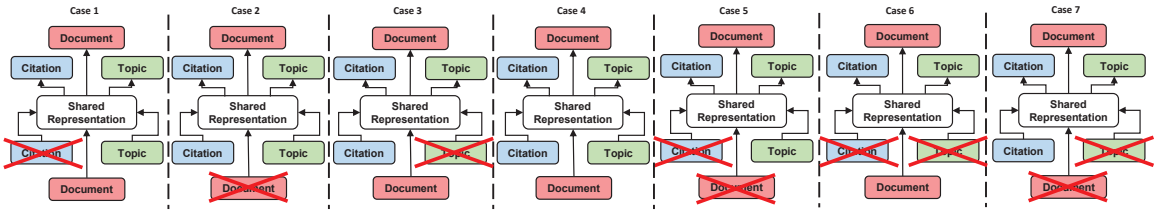


Figure 3.3: The process of imputation on multi-modal stacked autoencoder to deal with missing data.

resentation of multi-modal inputs; otherwise, missing inputs may yield poor shared representations. There are three possible cases of missingness (for different modalities):

1. Citation network representation: lacking citation information.

2. Document representation: missing abstract.

3. Topic representation: missing document representation as topic representation is computed by using document representation.

Ngiam *et al.* [85] proposed the use of an augmented dataset in the bimodal autoencoder that has a single modality as an input (the other input is set to zero values) that reconstructs two modalities as an output. However, naively extending this to the multi-modal scenario does not yield desirable results. We introduce a strategy that generalizes this work [85] for multi-modal in which we intentionally leave *one or more* representations out (or 'empty') while learning to induce a shared representation in a latent space from which we reconstruct all input representations. Figure 3.3 illustrates our proposed imputation process to construct the augmented dataset. In particular, for the inputs with no missing values, we purposely use an empty representation for each input and try to reconstruct the output with clean representations.

For illustration purposes, we demonstrate our process on a simple 2-dimensional example. Suppose we have 3 representations, $c = [1, 2]$, $d = [3, 4]$, and $t = [5, 6]$.

Then we train our encoder with all cases shown in Figure 3.3. For each case, the inputs are

- Case 1: $c = [0, 0]$, $d = [3, 4]$, and $t = [5, 6]$
- Case 2: $c = [1, 2]$, $d = [0, 0]$, and $t = [5, 6]$
- Case 3: $c = [1, 2]$, $d = [3, 4]$, and $t = [0, 0]$
- Case 4: $c = [1, 2]$, $d = [3, 4]$, and $t = [5, 6]$
- Case 5: $c = [0, 0]$, $d = [0, 0]$, and $t = [5, 6]$
- Case 6: $c = [0, 0]$, $d = [3, 4]$, and $t = [0, 0]$
- Case 7: $c = [1, 2]$, $d = [0, 0]$, and $t = [0, 0]$

and the reconstructed output is $c = [1, 2]$, $d = [3, 4]$, and $t = [5, 6]$ for all cases. Therefore, we intentionally leave one or two representations out using an *empty* representation (vector of zeroes) but still require the multi-modal autoencoder to reconstruct all representations. Using this process we can handle missing input representations because the model is forced to learn a robust shared representation from all possible combinations of the inputs.

3.2.3 Multi-label Classification Task

The objective of the MMiDaS-AE is to minimize the number of relevant articles (articles after the full-text screening) that are excluded while minimizing the number of irrelevant citations that need to be screened by domain experts. Thus, the model must make a binary prediction for each instance which indicates whether or not it should be screened by a human reviewer. Since SRs are intended to be *comprehensive* assessments of the relevant evidence, achieving high recall (i.e., sensitivity to the relevant citations) is imperative. This is challenging in practice because there is

severe *class imbalance* [53, 118], that is, there are far fewer relevant than irrelevant citations. Consider Figure 1.1: Here we have 2,544 articles in total, but only 183 (7.23%) and 41 (1.61%) of these pass abstract and full-text screening, respectively.

To ensure the identification of relevant articles, we propose a multi-label classification task to use the results of abstract screening as the labels are less imbalanced than the full-text. In the literature identification phase of SRs, there are two steps that are typically performed: title/abstract screening, which is followed by full-text screening. We posit that documents that pass the title/abstract screening are more likely to be "relevant" than those that are discarded. In other words, we were interested in ordering each document into three ordered categories: completely irrelevant, inclusion in the full-text screening, and inclusion in the SR. By including an abstract classifier, we can encode additional information that may help our model distinguish completely irrelevant articles. Thus, MMiDaS-AE uses two classifiers, an abstract classifier, and a full-text classifier. Then, as proposed by Niu *et al.* [87], we sum the prediction probability of the true (relevant) class for each classifier and use this to evaluate the performance of MMiDaS-AE. For example, if the article is predicted as irrelevant by the abstract classifier, it will have a low probability (and be unlikely to meet the final threshold). Thus, MMiDaS-AE will only detect articles that have high probabilities for both the abstract and full-text classifier.

Therefore, MMiDaS-AE consists of the following steps (as illustrated in Figure 3.2). We first train each feature representation, citation network, title/abstract, and MeSH terms into the citation, document, and topic representations. Then, we train a multi-modal stacked autoencoder to learn the shared representations that encode all three representations. While training the multi-modal stack autoencoder, we apply our proposed missing data imputation technique. Once the shared representation is learned, we use two softmax classifiers, an abstract classifier and a full-text classifier which is trained separately. The prediction probability of true (relevant) classes is

then the sum of these two classifiers.

3.3 Experimental Design

3.3.1 Data Preprocessing

For the experiment, we use the top 15 SRs and the last 4 SRs from Table 2.1. For extracting the metadata of each article, we use the PMID to extract the title, abstract, MeSH terms, and citations. In total, 37,149 unique articles were extracted using Entrez API[1]. The title and abstract of each article are concatenated together and pre-processed using the `nltk` library [11] in Python to remove stopwords, punctuations, and numbers. Each remaining word is then converted to a 200-dimensional vector representation using PMCVec[2] [34]. The individual word representations are then averaged to obtain the final document representation. Note that we also evaluated the results using a larger pre-trained language model, SciBert [10], and compared the results with PMCVec.

In the normal SR process, the initial list of articles is retrieved by the combination of MeSH terms. However, all the datasets do not contain the MeSH terms of each SR. Thus, we manually selected the MesH terms that describe each SR the best using the following process. For each SR, we obtained all the MeSH terms (information available from PubMed) that appear in the articles with their associated frequency. Then using the top 50 most frequent terms from this list, we manually searched and selected the MeSH terms that exist on the Wikipedia page associated with the topic (i.e., ACE Inhibitors). We also accounted for the number of times the term appears in the overall corpus to avoid "uninformative" terms such as "Humans", "Male", "Female", and "adult". After excluding terms that exist in the top 50 for all SRs,

[1] `https://www.ncbi.nlm.nih.gov/books/NBK25501/`

[2] Since PMCVec is pre-trained on PubMed abstracts, there was no case where a word did not have a vector representation.

each SR contains unique terms.

As introduced, we could not use the entire citation network because of the limitation of computational and memory footprint, thus we used a partial citation network[3]. Starting from the articles that are in the dataset, we looked backward and forward from the citation links by following the PMID to construct the citation network.

For learning the citation network representations using LINE, we used 3,158,195 vertices (articles) and 139,270,829 edges (citation links) which are extracted from all 19 SRs and their citations. For both first- and second-order proximity, we use 128 as the dimension of each representation, and as LINE [107] proposed, concatenated the first- and second-order proximity, resulting in 256 dimensions for the citation network representation.

3.3.2 Inter-topic Setting

As the Cohen dataset has 15 SR topics, we evaluate MMiDaS-AE with non-topic specific settings. Specifically, for model training, 14 SR topics are used to classify the one leftover SR topic to evaluate the workload saved. We compare the results with two existing works that used the same inter-topic settings.

- **Norman**: Norman *et al.* [88] constructs a ranker by extracting bag-of-n-grams in titles, abstracts using TF-IDF and binary features. Also, article metadata such as keywords, journal name, and publication types are used as features.

- **Cohen (2008)**: Cohen *et al.* [25] studies the performance of the Support Vector Machine (SVM) classifier using both textual (unigram and bigram terms of titles and abstract) and conceptual (MeSH terms) features.

[3]We attempted to construct higher-order citations but found that not only did crawling the network take time, but LINE did not converge within 2 days on a machine with 16 CPU cores and 100GB RAM.

3.3.3 Intra-topic Setting

Intra-topic is a topic-specific setting that only uses training data within the same topic. Intra-topic assumes that reviewers labeled small batches of articles. Previous works used 5×2 cross-validation within each SR topic to evaluate intra-topic. Under 5×2 cross-validation, each SR topic is divided into two parts – one split is used for training and the other for testing. Then the roles of each half are switched. This entire process is then repeated 5 times. 5×2 cross-validation results in 10 experiments and the final score is the average of the 10 experiment scores. We compare the results with four existing works that use the intra-topic setting.

- **Cohen (2006)**: Cohen *et al.* [26] uses a voting perceptron algorithm with varying learning weights using bag-of-words, MeSH terms, and publication type as their features.
- **Khabsa**: Khabsa *et al.* [57] uses textual features, co-citations, and brown clustering as features to train a random forest model.
- **Norman**: Norman *et al.* [88] uses the same method as an inter-topic setting but uses the intra-topic setting with 5×2 cross-validation.
- **Matwin**: Matwin *et al.* [75] uses similar features to Cohen *et al.* [26] but trained Complement Naive Bayes instead.

3.3.4 Fine-tuning Setting

As we target the inter-topic setting that learns a model to classify articles as a function of article-article relations and article-topic relations, we propose *fine-tuning* our pre-trained model to evaluate our model in the intra-topic setting. Under the *fine-tuning* setting, we follow the inter-topic setting to pre-train our model, then use an intra-topic setting (5×2 cross-validation) to fine-tune the weights of the pre-trained model.

For example, if we want to predict which articles are relevant for the "ACEInhibitors" SR, then we use the other 14 SR topics to pre-train MMiDaS-AE first, and then use one-half of the articles in "ACEInhibitors" to fine-tune the weights of the pre-trained MMiDaS-AE, reserving the other half for testing. Then the roles of each half are switched. In other words for each of the 10 intra-topic experiments, we use the same pre-trained MMiDaS-AE that was trained with 14 SR topics but is then fine-tuned on 50% of the topic-specific data. We repeat this procedure 5 times, as same as 5×2 cross-validation. We report the average estimated score across the 10 experiments.

3.3.5 Hyperparameter Tuning

We found empirically that using a unified length of the independent hidden layer performs better than other settings. The unified length of the independent hidden layer means learning all three representations into the same length. For example, we use a 256-dimensional vector representation for network representation and a 200-dimensional vector representation for document and topic representation. We add an independent hidden layer connected with the input representation with a 100-dimensional vector representation, thus after these layers, all three inputs will have equal dimensions. Also for the length of the shared representation (encoding dimensions), we empirically discovered that 50 works the best in our setting that balances the predictive power and the error in the reconstructed representation. For the activation functions in the multi-modal stacked autoencoder, we use Rectified Linear Units (ReLUs) for all encoders and the sigmoid activation function for all decoders.

Between 15 topics from the Cohen dataset, we left one topic out as the test set and used the other 14 topics as the training set. The other 4 datasets, COPD, proton beam, anemia, and clopidogrel, are used as a validation set to tune the hyperparameters and perform the testing on the topic that was being held out. For a fair comparison, we fixed the validation set to be these 4 datasets, so that SRs from the

Table 3.1: Comparison between MMiDaS-AE and other approaches in the inter-topic setting. Cohen *et al.* [25] only reported AUC, thus we only compared WSS@95% score with Norman *et al.* [88]. Bold scores are the top scores while underlined scores are the second-best scores.

Dataset	WSS@95%		AUC		
	MMiDaS-AE	Norman	MMiDaS-AE	Norman	Cohen (2008)
ACEInhibitors	**0.602**	0.566	**0.872**	0.817	0.806
ADHD	**0.661**	0.128	**0.727**	0.591	0.469
Antihistamines	**0.273**	0.073	**0.667**	0.652	0.62
AtypicalAntipsychotics	**0.244**	0.162	0.758	**0.759**	0.653
BetaBlockers	**0.445**	0.400	**0.850**	0.837	0.801
CalciumChannelBlockers	**0.381**	0.129	**0.894**	0.759	0.712
Estrogens	**0.256**	0.176	**0.705**	0.693	0.588
NSAIDS	0.654	**0.671**	0.901	**0.912**	0.899
Opioids	**0.678**	0.301	**0.885**	**0.885**	0.856
OralHypoglycemics	**0.115**	0.072	0.654	**0.657**	0.573
ProtonPumpInhibitors	**0.398**	0.377	**0.857**	0.823	0.793
SkeletalMuscleRelaxants	**0.502**	0.241	**0.848**	0.828	0.836
Statins	**0.341**	0.266	0.819	**0.826**	0.773
Triptans	**0.469**	0.464	**0.825**	0.819	0.823
UrinaryIncontinence	**0.451**	0.374	**0.895**	0.887	0.851

Cohen dataset are used only as training and test set. For the citation network, we used all articles in the partial citation networks (from all train, validation, and test sets), as LINE requires the entire graph as the input. For articles that are present on multiple topics, we remove the sample from the training to prevent data leakage and only use it for testing.

3.4 Empirical Results

In this section, we discuss the results from two different settings, inter-topic and fine-tuning. Then we evaluate variants of MMiDaS-AE using just one of the three features, different autoencoders (shallow versus stacked), and our proposed imputation method in the ablation study section.

3.4.1 Inter-topic Results

As MMiDaS-AE targets a general SR process where we do not assume that we have any labels of the topic, we first use the inter-topic setting. For the setting, we compare the obtained WSS@95% with the values of WSS@95% reported in existing approaches as the inter-topic setting. Table 3.1 summarizes the results of our model in the inter-topic setting. While Norman reported the scores for both WSS@95% and AUC, Cohen (2008) only reported the AUC score. Thus, we also computed the AUC score of MMiDaS-AE to be comparable with Cohen (2008).

As shown in the table, MMiDaS-AE outperforms Norman in WSS@95% in a range from 1% (Triptans) to 416% (ADHD) except for one SR (NSAIDS). Based on the WSS@95% scores, MMiDaS-AE reduces the reviewers' workload by 464 articles compared with Norman, which screens out 157 articles in CalciumChannelBlockers. For Opioids, MMiDaS-AE excludes 1,298 articles while Norman saves 576.

For AUC, MMiDaS-AE mostly outperforms other approaches. For the topics of AtypicalAntipsychotics, NSAIDS, OralHypoglycemics, and Statins, the AUC is lower than Norman but not by a substantial difference. This, coupled with the WSS@95% scores suggests that MMiDaS-AE may not perform as well on lower recall on these topics. Overall, the results show that with an inter-topic setting (non-topic specific setting) MMiDaS-AE performs well with a reasonable score. In other words, MMiDaS-AE works in a general case when we first start SR.

3.4.2 Fine-tuning and Intra-topic Results

While the inter-topic setting assumes that we do not have any labels for the SR topic, we can also assume that reviewers have labeled small batches of articles. To make this comparison, we use the fine-tuning setting and compare the results against other intra-topic approaches. As they report the score only in WSS@95%, we only compare our results in WSS@95% for this setting. The results are shown in Table 3.2.

Table 3.2: Comparison between MMiDaS-AE with fine-tuning setting and other approaches in an intra-topic setting that uses 5×2 cross-validation. The scores are in WSS@95%. Bold scores are the top scores while underlined scores are the second-best scores.

Dataset	MMiDaS-AE	Cohen (2006)	Khabsa	Norman	Matwin
ACEInhibitors	**0.693**	0.566	0.469	0.629	0.523
ADHD	0.674	**0.680**	0.447	0.616	0.622
Antihistamines	**0.287**	0.000	0.03	0.149	0.149
AtypicalAntipsychotics	**0.249**	0.141	0.199	0.21	0.206
BetaBlockers	**0.529**	0.284	0.361	0.511	0.367
CalciumChannelBlockers	**0.439**	0.122	0.287	0.398	0.234
Estrogens	0.262	0.183	0.18	0.292	**0.375**
NSAIDS	**0.671**	0.497	0.404	0.537	0.528
Opioids	**0.694**	0.133	0.455	0.590	0.554
OralHypoglycemics	**0.132**	0.090	0.074	0.111	0.085
ProtonPumpInhibitors	**0.431**	0.277	0.288	0.307	0.229
SkeletalMuscleRelaxants	**0.519**	0.000	0.371	0.429	0.265
Statins	**0.457**	0.247	0.400	0.436	0.315
Triptans	**0.485**	0.034	0.312	0.303	0.274
UrinaryIncontinence	**0.461**	0.261	0.411	0.422	0.296

For Antihistamines and SkeletalMuscleRelaxants, according to Cohen *et al.* [26], the classification process did not provide any savings, thus are marked as 0.000 in the "Cohen (2006)" column. Except for two SRs, ADHD and Estrogens, MMiDaS-AE outperforms other existing models. For ADHD, the size of the total articles as well as the list of articles that pass the full-text screening are small, thus, the fine-tuning process only marginally improves the results (0.661 in the inter-topic setting versus 0.674 in the fine-tuning setting). We also posit a similar issue with Estrogens, which is that the total number of articles is small, and thus fine-tuning only marginally helps. More notably, for Statins, MMiDaS-AE saves reviewers' workload by 1,583 articles while Cohen (2006) saves 856, Khabsa saves 1,386, and Matwin saves 1,091 articles.

Table 3.3: Ablation study on different settings. The scores are in WSS@95% using the inter-topic setting. Bold scores are the best scores.

Dataset	Shallow-AE	MMS-AE (Scibert)	MMS-AE (PMCVec)	Imput.
ACEInhibitors	0.196	0.325	0.430	**0.488**
ADHD	0.133	0.210	0.212	**0.297**
Antihistamines	0.077	0.214	0.243	**0.246**
AtypicalAntipsychotics	0.053	0.094	0.156	**0.171**
BetaBlockers	0.235	0.291	0.319	**0.377**
CalciumChannelBlockers	0.060	0.117	0.123	**0.152**
Estrogens	0.078	0.165	0.194	**0.217**
NSAIDS	0.208	0.523	0.528	**0.597**
Opioids	0.020	0.220	0.268	**0.379**
OralHypoglycemics	0.019	0.082	0.080	**0.108**
ProtonPumpInhibitors	0.178	0.336	0.315	**0.381**
SkeletalMuscleRelaxants	0.154	0.406	0.489	**0.495**
Statins	0.157	0.234	0.237	**0.292**
Triptans	0.199	0.278	0.330	**0.410**
UrinaryIncontinence	0.314	0.316	0.317	**0.323**

3.4.3 Ablation Study

In addition to the results for the two settings discussed, we evaluate the results achieved when we ablate the different components of MMiDaS-AE, summarized in Table 3.3 and Table 3.4. First, we use a basic autoencoder to compress each of the three representations, ("Document", "Topic", and "Citation") and only train on the individual representation. "Shallow-AE" concatenates the features of all three representations and passes it to a single auto-encoder which is then passed to a softmax layer. The "MMS-AE" is the multi-modal stacked autoencoder implementation [15] without any imputation. And finally, we show the results of our proposed imputation process. All the results are shown in Table 3.3 and Table 3.4 are only using binary classification with full-text screening as a label (not multi-label classification task) with an inter-topic setting. Therefore, the results are different from the results reported in Table 3.1 which also demonstrates the added benefit of using a multi-label classification task. Also to evaluate the results with a larger pre-trained language model, we compare the PMCVec representation with SciBert representation

Table 3.4: Ablation study on each component. The scores are in WSS@95% using the inter-topic setting. The results of the document, topic, and citation network representation using a basic autoencoder with a single input. Underlined scores are the best scores.

Dataset	Document (SciBert)	Document (PMCVec)	Topic	Citation
ACEInhibitors	<u>0.284</u>	0.128	0.080	0.104
ADHD	0.122	0.179	0.124	<u>0.283</u>
Antihistamines	0.097	0.097	0.090	<u>0.166</u>
AtypicalAntipsychotics	0.064	0.057	0.061	<u>0.119</u>
BetaBlockers	0.127	<u>0.279</u>	0.088	0.141
CalciumChannelBlockers	0.063	0.028	0.107	<u>0.137</u>
Estrogens	0.054	0.055	0.086	<u>0.158</u>
NSAIDS	0.168	0.077	0.226	<u>0.397</u>
Opioids	0.182	0.180	0.124	<u>0.232</u>
OralHypoglycemics	0.034	0.029	<u>0.082</u>	0.028
ProtonPumpInhibitors	0.182	0.175	0.032	<u>0.294</u>
SkeletalMuscleRelaxants	0.238	0.225	0.167	<u>0.364</u>
Statins	0.139	<u>0.174</u>	0.125	0.066
Triptans	<u>0.234</u>	0.102	0.216	0.204
UrinaryIncontinence	0.040	0.124	0.215	<u>0.273</u>

using the "Document" and "MMS-AE" settings. We only evaluated "Topic" using PMCVec as we also evaluated the result on "MMS-AE".

In comparing individual components in Table 3.3 and Table 3.4, if the test set has a large number of articles in total, it leads to a high WSS@95% when using the document representation only. For example, ACEInhibitors has 2,544 articles in total, and Statins has 3,465 articles in total, and both SRs have a relatively higher WSS@95% than other individual components. There are cases when using only the citation representation is better. This also depends on the number of articles that lack citation information. For example, ADHD has only 6% of articles missing citation information and Opioids have 8% of articles missing citation information, and both have higher WSS@95% for the citation representation than other individual components. However, for ACEInhibitors 17% of articles are missing citation information and Statins has 15% of articles missing citation information, thus both have a lower WSS@95% than using only document representation.

Learning a classifier for SR is a difficult task as we only use partial information (title, abstract, and MeSH terms) to predict whether the article passed the full-text screening (where full-text is not included as a feature). Our intuition is that citation network representation can complement the lack of full-text information to improve the overall performance as citations are used in the full-text. In SR, although reviewers consider texts (title, abstract, and full-text), this implicitly considers the co-citation information. By comparing "Citation" and "MMS-AE" in Table 3.3 and Table 3.4, we can see cases when WSS@95% of using only citation representation outperforms multi-modal settings such as ADHD and CalciumChannelBlockers. This demonstrates the usefulness of the citation information.

In most cases, Shallow-AE performs worse than individual components which implies that the simple concatenation of representations does not learn a robust shared representation that encodes all three representations. However, if we use MMS-AE, it performs better in all topics compared to Shallow-AE. This suggests that MMS-AE is learning a more robust shared representation than Shallow-AE. Finally, if we apply the imputation technique that we propose, it performs the best and can reduce the workload by up to 59.7% compared to the MMS-AE. In addition, a comparison between the WSS@95% scores in the "Imputation" column in Table 3.1 and Table 3.2 and the MMiDaS-AE column in Table 3.1, shows a significant improvement through the introduction of the multi-label formulation.

All the results shown in Table 3.1 and Table 3.2 are using PMCVec for the document and topic features. However, we wanted to evaluate the difference in using approaches that exploit pre-trained representations induced by large transformers such as SciBert [10]. We compared the results using SciBert and PMCVec on "Document" and "MMS-AE" in Table 3.3 and Table 3.4. As shown, for most of the cases when using a single-component (only document as a feature), SciBert performs better than PMCVec. However, when using the MMS-AE setting, PMCVec outperforms SciBert

in most of the cases. This illustrates the importance of the number of dimensions in MMS-AE. We note that the dimension of SciBert is 768 while the dimension of PMCVec is 200. When using a single-component, we can select the size of the hidden layer based on the input dimension. However, for MMS-AE, the three features are encoded into a shared representation, and it becomes difficult when the dimension of one input differs greatly from the other input. In other words, there will be information lost from the input feature with a larger dimension when learning the shared representation. Thus, in MMS-AE, information from the Document and Topic is lost when learning the shared representation and consequently performs worse than the single-component.

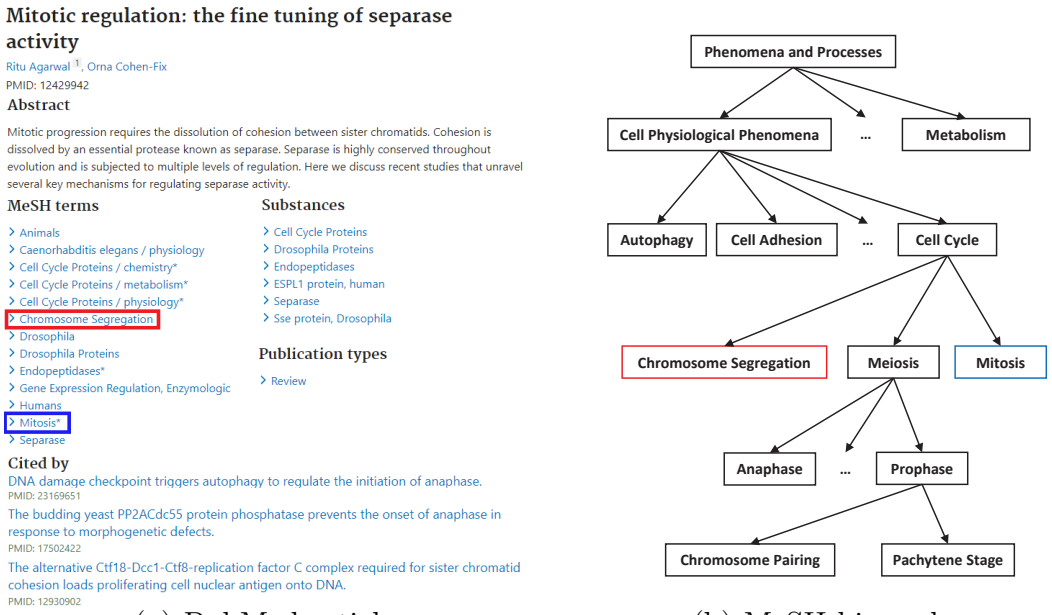
Chapter 4

PubMed Graph Benchmark

PubMed is a database that contains over 33 million citations and abstracts of literature related to biomedicine and health fields, as well as related disciplines such as life sciences, behavioral sciences, chemical sciences, and bioengineering [16]. PubMed articles have been used to perform numerous SRs [26, 51, 115], evaluate biological processes [63], identify protein-protein interactions [52], curate genes [124], and extract biological networks [129]. To date, much of the work on PubMed literature has focused on mining the text. The previous chapter illustrated that the citation structure can be utilized to automate the SR process, where for some topics, the node embedding provided better representations than their textual counterparts. Yet our work simplified PubMed to a homogeneous graph (only paper nodes and paper – paper edges).

Figure 4.1(a) shows an example of a Pubmed article[1]. In addition to the author, venue, and citation information that is commonly found in most bibliographic data, each PubMed article contains data regarding the Chemical Substances within the article, the type of article that characterizes the nature of the information or the type of research support received, and Medical Subject Headings (MeSH) terms which identify the broader concepts in the data. The categorical information of chemical

[1]Article can be found here`https://pubmed.ncbi.nlm.nih.gov/12429942/`.



(a) PubMed article (b) MeSH hierarchy

Figure 4.1: Example of PubMed Article (a) and the partial MeSH hierarchy (b) that is associated with the article. For article (a), the PubMed database contains pmid, title, abstract, list of articles cited by, publication types, list of MeSH terms, and list of substances (chemicals). The MeSh hierarchy (b) shows the categorization of the MeSH terms to a broader concept.

substances and publication types are not found in DBLP, ACM, or MAG. Moreover, there are over 30,000 terms in the MeSH vocabulary. Furthermore, the terms follow a hierarchical taxonomy[2] (see Figure 4.1(b) for an example MeSH tree for some terms in the example), yet also have the unique property that a term can belong to one or more trees[3]. Capturing this hierarchical structure can potentially improve the representation; however, the data in PubMed is incomplete as it does not perform author disambiguation.

We present **P**ubmed **G**raph **B**enchmark (PGB) [65], a new benchmark dataset for evaluating HIN embeddings for biomedical literature. PGB provides three different tasks to evaluate the quality of the network embeddings that span node classification, node clustering, and abstract screening for various SR tasks. The latter task is

[2] `https://www.nlm.nih.gov/mesh/intro_trees.html`

[3] We note ACM has the Computing Classification System (CCS) which is a hierarchical ontology but does not allow a term to belong to multiple trees.

Table 4.1: Comparison of existing bibliographic datasets with N denoting nodes, NT denoting node types, ET denoting edge types, and HIER denoting a hierarchical structure on at least one of the nodes.

Benchmark	# N	# NT	# ET	HIER
PGB	54,974,182	6	7	✓
ogbn-mag	1,939,743	4	4	✗
ogbn-arxiv	169,343	1	1	✗
ogbn-papers100M	111,059,956	1	1	✗
HGB-DBLP	26,128	4	6	✗
HGB-ACM	10,942	4	6	✗
HGB-Freebase	180,098	8	36	✗

different than the existing node-level and edge-level tasks provided in OGB and HGB in that the same node can have different labels depending on the SR content. By providing a high-quality and large-scale heterogeneous bibliographic network with three different graph tasks and their associated evaluation metrics, we can measure progress in a consistent and reproducible fashion.

4.1 Benchmark Comparison

Bibliographic data is used in various tasks, for example, word embedding using the title and abstract, network embedding using the citation, and author network. Thus, many works have worked on constructing a benchmark for bibliographic data such as OGB [50], HGB [73], and S2ORC [72]. Here we briefly describe the three related academic paper benchmark datasets and their limitations.

OGB [50] is a large-scale benchmark for graph machine learning tasks. It encompasses a variety of domains such as social networks, biological networks, molecular graphs, and knowledge graphs. OGB also has bibliographic data, for example, ogbn-arxiv and ogbn-papers100M are citation networks that are extracted from arxiv and MAG, respectively. Notably, both of these citation networks are homogeneous networks with paper nodes and links that represent the citation. Unlike ogbn-arxiv and

ogbn-papers100M, OGB also has a heterogeneous academic network, ogbn-mag, which is extracted from MAG. The ogbn-mag dataset contains 4 different node types (i.e., papers, authors, institutions, and topics) along with their relations. However, OGB mainly focuses on benchmarking graph machine learning methods on the large-scale homogeneous network.

HGB [73] provided eleven medium-scale graph benchmark datasets for node classification, link prediction, and knowledge-aware recommendation. For node classification, they provided DBLP, IMDB, ACM, and Freebase [12] datasets, and for link prediction, Amazon, LastFM, and PubMed datasets. The PubMed benchmark is the subset of a previously generated network of genes, diseases, chemicals, and species filtered by domain experts [129]. However, unlike PGB, the PubMed dataset does not reflect the bibliographic data directly. Instead, for HGB, DBLP, and ACM datasets serve as the lone benchmarks for the bibliographic network. contains paper, author, subject, and terms. However, both datasets lack metadata that can be helpful for learning node embeddings. Additionally, the benchmarks assume each node has a single label, whereas labels can change depending on the context.

The S2ORC [72] corpus is a large-scale academic paper corpus that is constructed using the data from the Semantic Scholar literature corpus [3]. Articles in Semantic Scholar are derived from numerous sources which are obtained directly from publishers such as MAG, arXiv, PubMed, and crawled from the open Internet. Semantic Scholar clusters these papers based on title similarity and DOI overlap, resulting in an initial set of approximately 200M paper clusters. Using the Semantic Scholar literature corpus, S2ORC aggregated the metadata of articles and cleaned the data to select canonical metadata using external sources such as IEEE and DBLP. Although S2ORC contains biomedical literature, it mainly focuses on the common metadata that exists across all the articles. Since publication types, MeSH terms, and chemical substances are only present in biomedical literature, such metadata is not in-

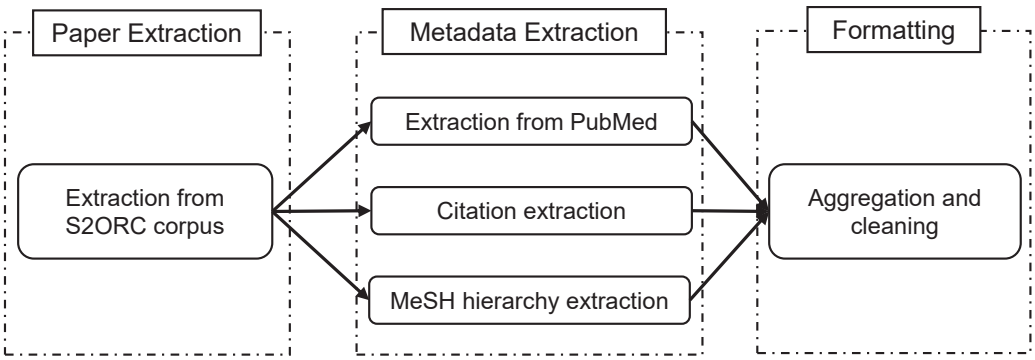


Figure 4.2: Framework overview of the PGB.

cluded in the dataset. Thus, developing embeddings that reflect the heterogeneity of the PubMed database requires additional preprocessing. Table 4.1 summarizes the statistics of the existing bibliographic datasets.

4.2 Benchmark Construction

In this section, we introduce the framework to construct PGB, shown in Figure 4.2.

4.2.1 Paper Collection

PGB is constructed based on the S2ORC corpus [72] as it contains more complete citation information than PubMed. However, there exist cases where the abstract only exists in the Semantic Scholar database but not in the PubMed database. Since PGB targets the biomedical literature, we initially extract articles that contain a PubMed ID (PMID) from S2ORC.

4.2.2 Metadata Extraction from PubMed

The S2ORC corpus only contains basic metadata of each PubMed paper (e.g., title, abstract, authors, year, and venue). In biomedical literature, unlike general academic articles, there is important metadata that can serve an important role such as Medical

Subject Headings (MeSH) terms and publication types. Additional partial information can be found in PubMed (see Figure 4.1). To extract more detailed information related to each article, we query information from the Entrez API[4] using the PMID.

The metadata contains "Chemical List", "Publication Type", and "MeSH Terms". The chemical list provides the registry number of specific chemical substances assigned by the Chemical Abstracts Service and the names of the chemical substances. The publication type identifies the type of article indexed for MEDLINE and characterizes the nature of the information, how it is conveyed, and the type of research support received. For example, an article can have a publication type of *Review*, *Letter*, *Retracted Publication*, *Research Support, N.I.H.*, or *Clinical Conference*. Finally, MeSH terms are used to characterize the content of the articles. MeSH terms are recorded with the information whether they are the major topic or not. The major MeSH terms denote that those are the most significant topics of the paper whereas the non-major MeSH terms are used to identify concepts that have also been discussed in the item but are not the primary topics. For the articles identified from our paper collection process, we integrate the names of the chemical substances, the publication type, and both major and minor MeSH terms.

4.2.3 Citation Extraction

While the PubMed database contains rich information on biomedical literature, it contains few information about the citations. However, S2ORC corpus extracted the citations from the collected PDF or LaTeX files on top of the Semantic Scholar literature corpus. Thus, to construct PGB, we first use the citation information from the S2ORC. This includes both in and out citations which refer to whether the paper is cited by another paper or the paper cites another paper. We convert all the Semantic Scholar IDs into PMIDs and remove papers that are not included in the

[4] `https://www.ncbi.nlm.nih.gov/books/NBK25501/`

PubMed database[5].

We note that there are cases where Semantic Scholar does not contain all the citations. Thus we also extract citations from the Entrez API to include papers that do not exist in the Semantic Scholar database but exist in PubMed. The metadata associated with the PMIDs of the newly identified papers is then retrieved to ensure consistency of the article information. In this fashion, the articles in PGB are not a pure subset of S2ORC.

4.2.4 MeSH Terms Hierarchy

One important feature of MeSH terms is the hierarchical ontology of the terms. MeSH terms can be categorized into broader MeSH terms that support the categorization of the articles, as depicted in Figure 4.1(b). The categories of different hierarchy levels reveal the similarity at coarser/fine-grained granularities. As shown in Figure 4.1(b), MeSH terms that are assigned to the article can share the same parents or can be in a different sub-tree. When comparing two articles, if they do not have the same MeSH terms but MeSH terms with the same parents (or within the same sub-tree), then the two articles are potentially closely related. Therefore, knowing the hierarchy can play an important role in identifying similar articles.

Unfortunately, the Entrez API does not include the MeSH terms hierarchy. Thus, we also extract the MeSH terms hierarchy dataset[6] to identify the position of the MeSH terms associated with each article. The MeSH terms hierarchy dataset only contains the MeSH terms shown in Figure 4.1(b). However, the tree numbers help reveal the hierarchical structure. For example, the MeSH terms "Chromosome Segregation" with tree number G04.144.220.220.625 and "Mitosis" with tree number G04.144.220.220.781 demonstrate that they share the same parent, "Cell Cycle" with

[5]While this can potentially harm or bias the embedding, we did this to maintain consistency in the article information in PGB.

[6]`https://www.nlm.nih.gov/databases/download/mesh.html`

Table 4.2: List of metadata included in the PGB, and the field name and the type in JSON format.

Field Description	Field name	Field type	Explanation
PMID	pmid	str	Pubmed ID
Title	title	str	Title of the paper
Abstract	abstract	str	Abstract of the paper
Authors	authors	List[Dict]	list of authors
Year	year	int	Published year
Venue	venue	str	Venue of the paper
Publication Type	publication_type	str	Type of the article
Chemical List	chemicals	List	Name of the chemical substances
MeSH Terms	mesh	List[Dict]	List of MeSH terms
In Citation Validator	has_inbound_citations	bool	Validator for inbound citation
Out Citation Validator	has_outbound_citations	bool	Validator for outbound citation
Inbound Citation	in_citation	List	List of PMID that cites the paper
Outbound Citation	out_citation	List	References of the paper

tree number G04.144.220.220. Thus, we integrate the tree number for each MeSH term using the MeSH terms hierarchy into PGB.

4.3 Data Format

Table 4.2 summarizes the field name and the field type that is used to store PGB. The "authors" field contains 4 subfields, "first", "middle", "last", and "suffix". For the chemical list and the MeSH terms, we exclude the ids and only included the name because the name itself is already unique. For the MeSH terms, we use 3 subfields to convey which MeSH terms are major or minor, The subfield "name" refers to the name of the MeSH terms, the subfield "is_major" is set to a true/false value to identify the major MeSH terms, and the subfield "tree_number" is the MeSH hierarchy information. There can exist multiple major MeSH terms for each article. We also included fields for validating whether the inbound and outbound citation exists in the benchmark. The fields are named "has_outbound_citations" and "has_inbound_citations", and the value is set to be either true or false. This helps users easily identify the presence of citation information without parsing the list of citations.

Figure 4.3: Statistics of PGB (30,872,730 articles).

Name	Total #	Missing (%)	Avg per article
Authors	30,397,681	1.54	4.11
Articles w/ MeSH terms	26,883,163	12.92	9.32
Articles w/ chemical list	14,565,380	52.82	1.82
Articles w/ publication type	30,685,975	0.60	1.73
Articles w/ inbound citations	16,488,646	46.59	6.51
Articles w/ outbound citations	7,781,767	74.79	6.51

4.3.1 Statistics

The statistics of PGB are shown in Table 4.3. It contains 30,872,730 biomedical literature, and all the articles have PMID, title, abstract, year, and venue. However, there exist cases in which any one of the fields is missing which denote that the information does not exist in both PubMed and S2ORC. 46.59% of articles do not have an inbound citation, and 74.79% of articles do not have outbound citations. Table 4.3 also shows the average number of MeSH terms, chemicals, and inbound and outbound citations. Due to the large size of the benchmark (∼60GB), PGB is split into 10 partitions where each partition is compressed as a zip file.

We can directly use the metadata in PGB to retrieve any necessary information to construct a homogeneous or heterogeneous network. There are 5 node types (Paper (P), Author (A), MeSH terms (M), Venue (V), and Publication type (T)) and 6 edge types (P – P, P – A, A – A, P – M, P – V, P – T). The MeSH hierarchy can be added using another edge type M – M. The constructed HIN can be used for node classification to determine the topic of articles, link prediction for citation recommendation, and SR for abstract/full-text screening. To illustrate the usage of PGB, we perform experiments using both homogeneous and heterogeneous network embedding models and evaluate the embedding for identifying articles for a SR, node classification, and node clustering.

```
{
   "pmid":"12429942",
   "pmcid":null,
   "title":"Mitotic regulation: the fine tuning of separase activity.",
   "abstract":"Mitotic progression requires the dissolution of cohesion between sister chromatids.
   Cohesion is dissolved by an essential protease known as separase. Separase is highly conserved
   throughout evolution and is subjected to multiple levels of regulation. Here we discuss recent
   studies that unravel several key mechanisms for regulating separase activity.",
   "authors":[
      {
         "first":"Ritu",
         "middle":"",
         "last":"Agarwal",
         "suffix":""
      },
      {
         "first":"Orna",
         "middle":"",
         "last":"Cohen-Fix",
         "suffix":""
      }
   ],
   "year":2002,
   "venue":"Cell cycle",
   "journal":"Cell cycle",
   "publication_type":[
      "Journal Article",
      "Review"
   ],
   "chemicals":[
      "Cell Cycle Proteins",
      "Drosophila Proteins",
      "Endopeptidases",
      "ESPL1 protein, human",
      "Separase",
      "Sse protein, Drosophila"
   ],
   "mesh":[
      {"term":"Animals","is_major":false,"tree_num":"B01.050"},
      {"term":"Caenorhabditis elegans","is_major":false,"tree_num":"B01.050.500.500.294.400.875.660.250.250"},
      {"term":"Cell Cycle Proteins","is_major":false,"tree_num":"D12.776.167"},
      {"term":"Chromosome Segregation","is_major":false,"tree_num":"G05.113.220.625"},
      {"term":"Drosophila","is_major":false,"tree_num":"B01.050.500.131.617.720.500.500.750.310.250"},
      {"term":"Drosophila Proteins","is_major":false,"tree_num":"D12.776.093.500.462"},
      {"term":"Endopeptidases","is_major":true,"tree_num":"D08.811.277.656.300"},
      {"term":"Gene Expression Regulation, Enzymologic","is_major":false,"tree_num":"G05.308.320"},
      {"term":"Humans","is_major":false,"tree_num":"B01.050.150.900.649.313.988.400.112.400.400"},
      {"term":"Mitosis","is_major":true,"tree_num":"G05.113.220.781"},
      {"term":"Separase","is_major":false,"tree_num":"D12.776.167.552"}
   ],
   "outbound_citations":[],
   "inbound_citations":["14625536","16207798","23169651"],
   "has_outbound_citations":false,
   "has_inbound_citations":true
}
```


Figure 4.4: Example of JSON format of the PGB associated with Figure 4.1.

4.3.2 Code and Data License Information

The entire data is released and publicly available on Zenodo.[7] Due to the size of the benchmark (∼60GB), we split the benchmark into 10 partitions and uploaded 10 zip files onto Zenodo. In the uncompressed file, each article is stored in JSON format as described in Figure 4.4.

PGB is released under the CC BY-NC 4.0 license and for non-commercial use. PGB is constructed using the PubMed Entrez API and S2ORC. S2ORC is non-

[7] `https://zenodo.org/record/6406776#.YqrOKnbMKUk`

commercial use and released under the same license (CC BY-NC 4.0 license). The PubMed Entrez API does not require a signed license agreement to download publicly accessible data. However, we note that the associated PubMed metadata (i.e., MeSH terms, Chemical list, and publication type) in PGB may not reflect the most current data available on PubMed. The data can be re-updated using the Github repository assuming no major changes in the type and format of the machine-readable data. The usage guidelines and registration for the API key are detailed in the electronic book chapter[8]. Note that potential publication bias or other ethical considerations may need to be considered further.

4.4 Experimental Design

4.4.1 Data Preprocessing

We evaluate our model on the popular, and publicly available 15 SRs provided by Cohen *et al.* [26], 3 SRs provided by SWIFT-Review [49], and 3 SRs from CLEF-TAR [55] dataset. The detailed information is described in Table 2.1. Each SR topic contains a set of articles that were retrieved using specific search queries that combined the health condition and the drug intervention. Each article, identified using a PubMed identifier (PMID), was triaged using a two-step process. First, the abstract is reviewed to determine if it meets the inclusion criteria of the SR. If the criteria are met, the entire article is then reviewed to determine if the evidence should be summarized in the SR. We target the abstract screening process where most of the articles are excluded. Note that the same article can be included in one or more SR topics and have different abstract triage status, unlike existing tasks in OGB or HGB.

[8] `https://www.ncbi.nlm.nih.gov/books/NBK25497/#chapter2.Usage_Guidelines_and_Requiremen`

4.4.2 Baseline Models

We evaluate six different models that include document embedding, homogeneous network embedding, HIN embedding, and knowledge graph embedding models.

- SPECTER [24]: The SPECTER is an embedding model that learns the representation of a document by computing the embeddings using a SciBERT model [10] pre-trained on relatedness signals derived from the citation graph. We use the embeddings for SPECTER provided by Semantic Scholar API.[9] The Semantic Scholar API allows a paper search by using the PubMed ID to retrieve the Semantic Scholar ID. Using this Semantic Scholar ID, we retrieve the SPECTER embeddings of each document.

- LINE [107]: LINE is a conventional homogeneous network embedding method that uses first- and second-proximity. LINE uses the joint probability between two nodes.[10] We set the number of dimensions to 128 for both first- and second-proximity. The final embedding is the concatenation of 2 proximities. As LINE is an unsupervised model, we add a soft-max layer on top of the final embeddings.

- GCN [59]: GCN is a graph convolutional network embedding model designed for a homogeneous network.[11] GCN is trained in a supervised setting using the SR task. We use the 500-dimension TF-IDF weighted word vector provided by Namata *et al.* [82] as the node feature.

- HAKE [136]: HAKE is a hierarchical-aware knowledge graph embedding model which is not a GNN-based model but a translational distance model which describes relations as translations from one node to the other.[12] It uses radial

[9] `https://www.semanticscholar.org/product/api`
[10] `https://github.com/DeepGraphLearning/graphvite`.
[11] `https://github.com/tkipf/gcn`.
[12] `https://github.com/MIRALab-USTC/KGE-HAKE`.

coordinates to embed entities at different levels of the hierarchy and uses angular coordinates to distinguish entities at the same level of the hierarchy. HAKE uses the link-prediction task to learn the embeddings, and thus it is an unsupervised model. For the supervised tasks, we add a soft-max layer on top of the embeddings. We try [500, 1000] for the dimension size and select 1000 as the validated parameter.

- GAHNE [70]: GAHNE is a model to learn representations for HIN.[13] It converts the network into a series of homogeneous sub-networks to capture semantic information. An aggregation mechanism then fuses the sub-networks with supplemental information from the whole network. Using the validation set, we process a grid search using [0.01, 0.005, 0.001] for the learning rate, [0.0005, 0.001] for the L2 penalty, and [64, 128, 256] for the dimension size. The validated parameters we used are a learning rate of 0.005, a dropout of 0.5, an L2 penalty of 0.001, and a dimension of 128. GAHNE is a supervised model and the model is trained using the labels from SR topics.

- ie-HGCN [132]: ie-HGCN is a GCN-based HIN embedding model that evaluates all possible meta-paths and projects the representations of different types of neighbor objects into a common semantic space using object- and type-level aggregation.[14] We use the supervised, cross-entropy loss to learn the weights of the model. We set the number of layers to be 5 and using the validation set, we tried [(128, 64, 32, 16), (156, 128, 64, 32)] as the dimension size. The validated parameters used in the results are 5 layers, with the dimensions of input, 128, 64, 32, and 16. We use the same node feature as GCN.

LINE and HAKE are unsupervised models, and the other 3 GNN baselines are semi-supervised models. For the SR task, for the homogeneous network, we only use

[13] `https:github.com/seanlxh/GAHNE`.
[14] `https://github.com/kepsail/ie-HGCN/`.

the citation information to construct the network. For HAKE, we use three node types (Paper (P), MeSH terms (M), Publication type (T)) with four edge types (P – P, P – M, M – M, P – T). For the other two HIN embedding models, we use four node types (Paper (P), Venue (V), MeSH terms (M), Publication type (T)) with four edge types (P – P, P – V, P – M, and P – T)

We use the code and perform a parameter search around the neighborhood of suggested parameters provided by the original paper. For each of the implementations, we kept separate environment files to ensure that the required Python packages were installed and the correct version as outlined in the code. The validation set is used to tune the hyperparameters for GAHNE and ie-HGCN.

4.4.3 Experimental Setup

For the SR task, we construct 3 different subsets of PGB for computational reasons. We trace the inbound and outbound citations up to 2-hops from the original articles to construct 3 subnetworks of approximately 1.2M, 3.4M, and 1.8M articles, respectively for the Cohen, SWIFT-Review, and CLEF-TAR datasets. In each subnetwork, we randomly split the graph into train-validation-test by sampling articles within each SR task using a 50%-25%-25% ratio. We create 3 train-validation-test trials for each subnetwork. For all the baselines, we used a g4dn AWS instance with NVIDIA T4 GPU.

4.5 Empirical Result

All the results shown in this section use the subnetwork of each dataset (Cohen, SWIFT-Review, and CLEF-TAR). We compare the performance of 1 language model and 5 network embedding models on SR. The performance is reported in Table 4.3 in the average of AUC scores of 3 trials for each SR task and in Table 4.4 in the average

Table 4.3: SR statistics and average AUC results across the three trials for the various models. The best score is bolded and the second highest is underlined.

Dataset	SPECTER	LINE	GCN	HAKE	GAHNE	ie-HGCN
Cohen-ACEInhibitors	0.677	0.580	0.592	0.677	<u>0.731</u>	**0.740**
Cohen-ADHD	0.567	0.548	0.577	0.599	<u>0.600</u>	**0.607**
Cohen-Antihistamines	0.505	0.493	0.509	0.521	**0.558**	<u>0.542</u>
Cohen-AtypicalAntipsychotics	0.638	0.555	0.597	0.648	**0.708**	<u>0.699</u>
Cohen-BetaBlockers	0.699	0.586	0.606	0.683	**0.733**	<u>0.728</u>
Cohen-CalciumChannelBlockers	0.601	0.594	0.608	0.621	**0.654**	<u>0.651</u>
Cohen-Estrogens	0.637	0.544	0.588	0.647	**0.676**	<u>0.673</u>
Cohen-NSAIDS	0.694	0.586	0.615	0.690	**0.767**	<u>0.746</u>
Cohen-Opioids	0.675	0.603	0.637	0.686	<u>0.725</u>	**0.727**
Cohen-OralHypoglycemics	0.535	0.512	0.529	0.533	**0.567**	<u>0.557</u>
Cohen-ProtonPumpInhibitors	0.674	0.604	0.626	0.681	**0.731**	<u>0.729</u>
Cohen-SkeletalMuscleRelaxants	0.688	0.605	0.632	0.687	<u>0.724</u>	**0.733**
Cohen-Statins	0.668	0.572	0.608	0.662	<u>0.710</u>	**0.716**
Cohen-Triptans	0.658	0.587	0.618	0.668	**0.723**	<u>0.717</u>
Cohen-UrinaryIncontinence	0.696	0.605	0.633	0.681	**0.745**	<u>0.741</u>
SWIFT-Transgenerational	0.695	0.637	0.667	0.684	<u>0.741</u>	**0.761**
SWIFT-PFOS-PFOA	0.671	0.634	0.657	0.695	<u>0.721</u>	**0.728**
SWIFT-BPA	0.632	0.563	0.604	0.645	<u>0.725</u>	**0.729**
CLEF-Prognosis-CD012661	0.678	0.593	0.628	0.647	<u>0.671</u>	**0.691**
CLEF-DTA-CD008803	0.619	0.598	0.628	0.643	<u>0.681</u>	**0.691**
CLEF-Intervention-CD005139	0.665	0.623	0.646	0.666	<u>0.702</u>	**0.704**

of WSS scores with the same setting. The best results are bolded and the second-best results are underlined.

As shown in the tables, both of the results (AUC and WSS scores) of the heterogeneous network embedding models (HAKE, GAHNE, and ie-HGCN) significantly outperform the homogeneous network embedding models (LINE and GCN). This suggests that not only the citation information but also other node types (venue, MeSH terms, and publication type) help to improve the performance of the SR task. GAHNE and ie-HGCN outperform HAKE as HAKE is an unsupervised model while others are semi-supervised models. However, by comparing the performance with the homogeneous model, HAKE shows the importance of the hierarchical information (MeSH hierarchy). The performance between GAHNE and ie-HGCN is similar. The results suggest that ie-HGCN performs better when there are more articles excluded from the

Table 4.4: SR statistics and average WSS results across the 3 trials for the various models. The best score is bolded and the second highest is underlined.

Dataset	SPECTER	LINE	GCN	HAKE	GAHNE	ie-HGCN
Cohen-ACEInhibitors	0.388	0.343	0.364	0.385	<u>0.472</u>	**0.489**
Cohen-ADHD	0.274	0.247	0.253	0.277	<u>0.343</u>	**0.344**
Cohen-Antihistamines	0.111	0.042	0.079	0.109	**0.168**	<u>0.137</u>
Cohen-AtypicalAntipsychotics	0.092	0.059	0.066	0.087	**0.111**	<u>0.102</u>
Cohen-BetaBlockers	0.209	0.186	0.19	0.211	<u>0.291</u>	**0.304**
Cohen-CalciumChannelBlockers	0.21	0.173	0.194	0.208	<u>0.221</u>	**0.242**
Cohen-Estrogens	0.223	0.169	0.197	0.222	**0.259**	<u>0.256</u>
Cohen-NSAIDS	0.385	0.377	0.384	0.437	**0.508**	<u>0.505</u>
Cohen-Opioids	0.253	0.21	0.218	0.276	<u>0.339</u>	**0.343**
Cohen-OralHypoglycemics	0.111	0.057	0.065	0.102	**0.133**	<u>0.128</u>
Cohen-ProtonPumpInhibitors	0.233	0.194	0.204	0.249	**0.287**	<u>0.283</u>
Cohen-SkeletalMuscleRelaxants	0.198	0.143	0.165	0.204	<u>0.239</u>	**0.246**
Cohen-Statins	0.229	0.169	0.179	0.227	<u>0.255</u>	**0.256**
Cohen-Triptans	0.343	0.278	0.294	0.348	**0.372**	<u>0.362</u>
Cohen-UrinaryIncontinence	0.21	0.162	0.174	0.202	**0.233**	<u>0.232</u>
SWIFT-Transgenerational	0.202	0.111	0.155	0.191	<u>0.253</u>	**0.277**
SWIFT-PFOS-PFOA	0.241	0.195	0.203	0.258	<u>0.378</u>	**0.383**
SWIFT-BPA	0.354	0.258	0.287	<u>0.376</u>	**0.441**	**0.441**
CLEF-Prognosis-CD012661	0.207	0.152	0.164	0.205	**0.252**	<u>0.248</u>
CLEF-DTA-CD008803	0.302	0.219	0.222	0.297	**0.341**	<u>0.337</u>
CLEF-Intervention-CD005139	0.2	0.143	0.158	0.199	<u>0.278</u>	**0.283**

abstract screening phase. For example, the "SWIFT-BPA" dataset has a total of 7700 articles in the beginning but only 111 articles (1.44%) are selected. Whereas ie-HGCN performs better in cases when fewer articles are selected, GAHNE performs better in cases when more papers are selected. For example, "Cohen-AtypicalAntipsychotics" starts with 1120 articles, and 363 articles (32%) passed the screening.

By comparing with the language model (SPECTER), it shows similar results with HAKE. In other words, SPECTER outperforms the homogeneous network embedding models (LINE and GCN) which only uses the citation network but underperforms the heterogeneous network embedding models (GAHNE and ie-HGCN). Although SPECTER is based on the transformer language model, it uses the document-level relatedness from the citation graph. Thus, this helps SPECTER to outperform the supervised homogeneous network embedding models. This illustrates the importance

of both the abstract and the citation graph in the SR process. Yet, even integrating the text and citation together does not beat the rich contextual information found in the venue, MeSH terms, and publication type.

Chapter 5

Community Multi-view based Enhanced Graph Convolutional Network

Given the rich metadata in PGB, we propose SR-CoMbEr [66], a **Co**mmunity **M**ulti-view **B**ased **E**nhanced GCN for **S**ystematic **R**eview to integrate the various node and edge types. SR-CoMbEr extends the multiple local GCN approach [123] to the HIN setting while also introducing a technique to eliminate additional hyperparameter settings. To automatically learn from the different object and link types, SR-CoMbEr adopts a multi-view approach at the community level to learn a view-specific embedding representation associated with each community. As a result, each community representation can capture the complex structure information across different relation types. Moreover, the multiple community GCN aggregation problem is posed as a multi-modal problem to yield a robust final embedding that reflects the different community representations. Our main contributions to this work are:

- We pose the problem of HIN representation as a multi-view learning problem to avoid specification of the meta-path while automatically capturing the network

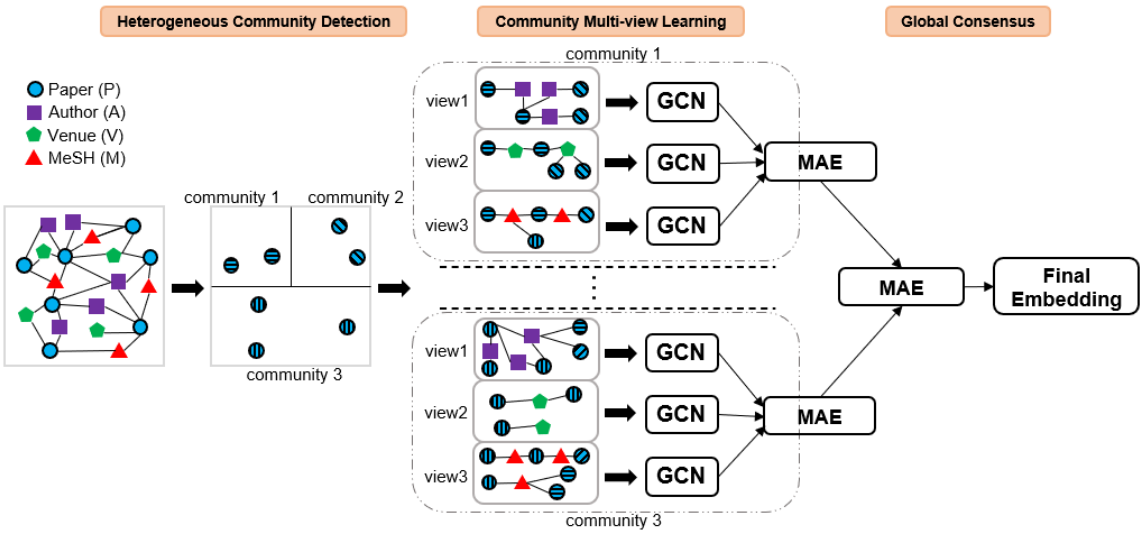


Figure 5.1: The framework overview of the SR-CoMbEr. The input network is a toy example of a PubMed Network which contains four node types and three edge types. Four node types are Paper (P), Author (A), Venue (V), and MeSH Terms (M), and three edge types are P – A, P – V, and P – M. The target node is set to P which is used for the node classification task.

semantics.

- We propose an innovative multiple, community-based multi-view GCN to capture the structural heterogeneity that is useful for downstream tasks.

- We conduct extensive experiments on SR screening to demonstrate the superior performance of SR-CoMbEr over HIN baselines.

5.1 Model Design

SR-CoMbEr is inspired by the multiple-filtering local GCN model [123], which constructs multiple local versions of a homogeneous network to capture different aspects of the node attributes while providing robustness to noise. Yet, the local versions of the multiple GCN approach may fail to capture the complex neighborhood structure when solely focusing on a homogeneous network. Moreover, the model can be sensitive to the number of local filters. We address these limitations using three parts: (1)

automatic identification of communities in HIN, (2) community multi-view learning to capture information from each link type, and (3) global consensus across the communities. Figure 5.1 depicts SR-CoMbEr's overall architecture, where the goal is to learn the representation of the target object α (i.e., circle node (P)).

5.1.1 Heterogeneous Community Detection

The ability to capture the neighborhood information is a crucial aspect of ensuring the quality of the network embedding. Many network embedding methods use random walks to capture the neighborhoods before passing them to a deep learning model. For example, the multiple-filtering local GCN model [123] uses random walk to construct $\mathcal{M}$ local networks are constructed. However, sampling of a single link type may not encapsulate the community structure via other link types while sampling multiple links may not be sufficient to capture the complicated structure [134]. However, utilizing the entire HIN can pose computational problems for large networks as well as limit their generalizability to unseen data [123]. Instead, we propose to utilize the community structure ubiquitous in networks, where a group of nodes exhibits more intra-connections than inter-connections with external nodes [35], to determine the construction of the local networks. Given a set of communities, a random walk is initiated using the nodes belonging to the community. Thus each local GCN version learns a better local embedding by integrating information found in the community structure. It is important to note that SR-CoMbEr does not restrict the random walk to just links between community nodes, therefore the local network may contain neighborhood information of nodes outside the community. Moreover, since a node may be part of multiple communities, the combination of multiple local GCNs will thereby reflect different neighborhood information for the same object.

The community-based focus of each local GCN lends itself naturally to automatic detection of the "optimal" number of local filters, $\mathcal{M}$. While there are many

types of community detection methods including clustering-based methods [83] and modularity-based methods [84], many of these models are developed for the homogeneous setting. Instead, SR-CoMbEr uses Tucker decomposition [111], a popular tensor factorization model, to identify the community structure and the number of optimal filters in the HIN setting. Tucker decomposition can be viewed as a generalization of singular value decomposition (SVD) which can detect communities in homogeneous networks [99]. The HIN tensor, $\mathcal{X}$, is a higher-order tensor where each object type serves as a mode of the tensor and the entries in the tensor capture the status of the links between the different modes of the tensor. For Figure 5.1, a paper by author by venue by term tensor (4-mode tensor), can be constructed where each element captures who authored a paper, where it was published, and what terms were present in that paper. Thus, the tensor succinctly encapsulates the relations between different object types.

Formally, for a 3$^{\text{rd}}$ order tensor, $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, Tucker decomposition approximates the tensor into a core tensor, $\mathcal{H} \in \mathbb{R}^{P \times Q \times S}$ multiplied by a factor matrices along each mode, $A \in \mathbb{R}^{I \times P}, B \in \mathbb{R}^{J \times Q}, C \in \mathbb{R}^{K \times S}$:

$$\mathcal{X} \approx \mathcal{H} \times_1 A \times_2 B \times_3 C. \tag{5.1}$$

The core tensor, $\mathcal{H}$ captures the level of interactions between the different components, and the factor matrices, A, B, C, are often assumed to be column-wise orthonormal. We note that Tucker decomposition generalizes to any N-mode tensor, does not impose column-wise orthonormal factor matrices nor does the core tensor have a decreasing Frobenius norm along each matrix slice. In addition, the column rank of each factor matrix can be different (i.e., $P \neq Q \neq S$) in the Tucker decomposition. We refer the reader to [60, 92] for additional details.

Since each local filter encapsulates a community, the column rank of each factor

matrix is set to be the same, $P = Q = S$. To compute the Tucker decomposition, we use the higher-order orthogonal iteration (HOOI) algorithm as it is one of the more efficient techniques. HOOI uses SVD to compute the orthonormal basis of each factor matrix [28]. Moreover, the resulting core tensor and factor matrices can be seen as the generalized counterparts of the matrix SVD. Thus, the superdiagonal entries of the core tensor ($H_{iii}, \forall i \in [1, R]$) are comparable to the singular values of SVD (i.e., diagonal entries in Σ). As a result, the number of communities can be calculated as the point in which the superdiagonal values converge, similar in fashion to using the Σ matrix in SVD to find the number of communities in a homogeneous network [99]. This eliminates the need for the user to grid search the number of filters $\mathcal{M}$.

The next step is to identify the nodes that belong to each community. Without loss of generality, we assume that the target object, α, corresponds to the first mode of the tensor. Each object can then be represented in a low-dimensional vector space (i.e., $P << I$) using the row vectors of the corresponding factor matrix A. Spectral clustering is performed on A to identify the community members using $\mathcal{M}$ for each node in the target object α. For simplicity of implementation, SR-CoMbEr uses the k-means algorithm to generate a hard cluster assignment but the framework can use any spectral clustering method. The graph for each community (local) filter is then obtained by performing a fixed-size random walk starting with only nodes within the community. Note that the community filters can contain not just nodes within the same community but also other nodes that are connected during the random walk process. The entire community detection process is summarized in Algorithm 1.

5.1.2 Community Multi-view Learning

Since random walk of $\mathcal{G}$ directly may fail to capture the complex structure, SR-CoMbEr treats each link type as a different view of the network. For each link type containing the target object α, a view of the community is created by performing

Algorithm 1: Heterogeneous Community Detection in SR-CoMbEr.

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi, \psi)$, $\phi : \mathcal{V} \to \mathcal{A}$, $\psi : \mathcal{E} \to \mathcal{R}$
Output: Number of filters $\mathcal{M}$, Communities $C_1, ..., C_M$

1 Construct tensor $\mathcal{X}$ from $\mathcal{G}$;
2 Compute $\mathcal{X} \approx \mathcal{H} \times_1 A \times_2 B \times_3 C$ using HOOI;
3 Set $\mathcal{M}$ based on convergence of values in the superdiagonal entries of $\mathcal{H}$;
4 Detect communities of α, $C_1, C_2, \ldots, C_M$, using spectral clustering of A;
5 **return** $\mathcal{M}$, $C_1, C_2, \ldots, C_M$;

the fixed-size random walk using only this link type. For each community GCN m, a view is constructed from each link type thus yielding $|\mathcal{R}|$ different representations, $X_1^m, \cdots, X_{|\mathcal{R}|}^m$. As an example, three views are constructed for Figure 5.1 with a different link type (e.g., P – A, P – V, P – M). Thus, rather than having a single community GCN, each community will have multiple view-specific filters of the network.

Although each view contains a single link type, GCN still cannot be applied directly because the neighbors of an object are of different types. Moreover, the adjacency matrix is not a square matrix and thus cannot be fed into Equation (2.1), where $\tilde{A}$ is the square matrix. We thus use the idea of projection, introduced in ie-HGCN [132], to ensure both object types are in the same space. Suppose the view captures the link α–β, where $\mathcal{V}^\alpha$ and $\mathcal{V}^\beta$ represent the set of objects in the α and β node type, respectively. Let $A^{\alpha-\beta} \in \boldsymbol{R}^{|\mathcal{V}^\alpha| \times |\mathcal{V}^\beta|}$ denote the adjacency matrix between α and β and the degree matrix $D^{\alpha-\beta} = diag(\sum_j A_{ij}^{\alpha-\beta}) \in \boldsymbol{R}^{|\mathcal{V}^\alpha| \times |\mathcal{V}^\alpha|}$. Every object is then projected into the same space and passed to the GCN:

$$\begin{aligned} \tilde{A}^{\alpha-\beta} &= (D^{\alpha-\beta})^{-1} \cdot A^{\alpha-\beta} \\ X_{\alpha-\beta} &= \tilde{A}^{\alpha-\beta} \cdot W^{\alpha-\beta} \end{aligned} \tag{5.2}$$

where $\tilde{A}^{\alpha-\beta}$ is the row-normalized matrix and $W^{\alpha-\beta}$ is the trainable convolution weight matrix of α–β relation.

The community embedding, X^m, should capture all the information from the $|\mathcal{R}|$ views while reducing information redundancy that may be present in the views. Moreover, certain views may learn better representations of the community. Thus, to summarize the different view modalities simultaneously, SR-CoMbEr adopts the multi-modal stacked autoencoder (MAE) [15]. MAE takes multiple input representations, concatenates the input together, and then passes this to an autoencoder to induce a succinct, shared representation from which to reconstruct the original (concatenated) inputs. Formally, the global consensus process for the shared representation in the m^{th} community GCN is:

$$H^m = MAE(X_1^m, X_2^m, \ldots, X_{|\mathcal{R}|}^m). \tag{5.3}$$

5.1.3 Global Consensus

Since each community multi-view GCN representation H^m, captures community-specific information, the learned representation can differ. We formulate the aggregation of the community multi-view GCN representation as a multi-modal problem. Although the final shared representation can be computed as the average of the community representations, this assumes each community is equivalent. In practice, some community representations are of higher quality and thereby should have higher weights. MAE is used again to learn the final representation across the $\mathcal{M}$ communities:

$$H = MAE(H^1, H^2, \ldots, H^{\mathcal{M}}) \tag{5.4}$$

The final embedding representation, H, is then used for a variety of tasks such as classification, clustering, etc, where the loss function is tailored towards the specific

Algorithm 2: The pseudocode of SR-CoMbEr.

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi, \psi)$, $\phi : \mathcal{V} \to \mathcal{A}$, $\psi : \mathcal{E} \to \mathcal{R}$
Number of localized filters $\mathcal{M}$
Output: Final representation H

Compute $\mathcal{M}, C_1, C_2, \ldots, C_M$ using Algorithm 1;
`/* Loop through the communities */`
for *i=1, ...,* $\mathcal{M}$ **do**
  `/* Loop through the views */`
  for $\alpha - \beta \in \mathcal{R}$ **do**
    Run random walk on objects $\in \mathcal{C}_i^{\alpha}$ and $\in \mathcal{C}_i^{\beta}$;
    Compute $X_{\alpha-\beta}$ according to Eq. (5.2);
  end
  Compute H^i according to Eq. (5.3);
end
Compute loss and update parameters;
return H according to Eq. (5.4);

task. For example, in a multi-class node classification task, H is passed to a fully connected layer with softmax activation, and the loss is defined as the cross-entropy over the object type. The weights are then learned using stochastic gradient descent with backpropagation. Algorithm 2 shows the overall training procedure of SR-CoMbEr.

5.2 Experimental Design

5.2.1 Data Preprocessing

We evaluate our model on the publicly available dataset provided by Cohen *et al.* [26]. The detailed information is described in Table 2.1. The dataset includes 15 SRs (or topics) concerning different drug efficacies that were performed by members of evidence-based practice centers (EPCs). In the dataset, each SR topic contains a set of PubMed article identifiers (PMID) and their associated title/abstract screening status (i.e., whether or not the article passed the title/abstract screening stage). The PMID allows us to retrieve the metadata (citation, author, venue, and MeSH terms)

Table 5.1: Comparison of baseline characteristics. The * symbol next to the model name denotes a homogeneous network model. The columns MP, SS, and MVF represent meta-path specification, subgraph sampling, and multi-view fusion, respectively.

	MP	SS	MVF	Module	Supervision
LINE*	✗	✗	✗	Skip-gram	✗
GCN*	✗	✗	✗	GCN	✓
HAN	✓	✗	✗	Transformer	✓
GAHNE	✗	✓	✓	GCN	✓
ie-HGCN	✗	✓	✓	GCN	✓
SR-CoMbEr	✗	✓	✓	GCN	✓

from the PubMed database. There exist other SR datasets [101], however, the dataset does not contain the PMID. We extract a subset of articles from the PubMed database using Entrez API[1]. Including all the articles from the Cohen dataset and using Entrez API, we trace articles up to 3-hops based on the citation information and retrieve about 7.6M articles with the meta-data including author, venue, and MeSH terms.

5.2.2 Baseline Models

We compare with five baselines spanning both homogeneous and HIN embedding methods in the SR task. Table 5.1 compares the characteristics of baseline models.

- **LINE [107].** A conventional network embedding method that uses first- and second-proximity. Since it is designed for a homogeneous network, we transform the HIN by collapsing the object and link types as a single type and using LINE to learn the representation of the whole HIN.

- **GCN [59].** A semi-supervised graph convolutional network that is designed for a homogeneous network. Similar to LINE, we ignore the heterogeneity of the network and collapse it into a homogeneous network.

- **HAN [122].** A model to learn representations for HIN. It transforms the HIN

[1] `https://www.ncbi.nlm.nih.gov/books/NBK25501/`

into several homogeneous sub-networks by user-defined meta-paths. For object-level aggregation, it uses GAT [112], then uses an attention mechanism to fuse object representations from each sub-network.

- **GAHNE [70].** A model to learn representations for HIN. It converts the network into a series of homogeneous sub-networks to capture the semantic information. Then an aggregation mechanism fuses the sub-networks with supplemental information from the whole network.

- **ie-HGCN [132].** A HIN embedding model that evaluates all possible meta-paths and projects the representations of different types of neighbor objects into a common semantic space using object- and type-level aggregation.

5.2.3 Implementation Details

The source codes of the other baselines are provided by their authors and are implemented in either PyTorch or TensorFlow. All experiments are conducted on a machine with 1 Nvidia GeForce GTX 1080Ti and 11GB GPU memory. For each SR task, we randomly split the articles in the SR into train-validation-test as 50%-25%-25%, and use the validation set for the hyperparameter tuning. Articles not in the target SR task are marked as irrelevant in the training process.

For the baseline models, we adopt the same hyperparameter settings introduced in their respective papers. For LINE [107], we use a dimension of 128 for each first- and second-order proximity resulting in a dimension of 256 for the final embedding. For GCN [59], we use the learning rate of 0.01, the dropout rate of 0.5, and the L2 penalty weight decay of 0.0005. For HAN [122], the number of attention heads is set to 8, and the meta-paths PAP, PMP, and APVPA are used (P: Paper, A: Author, M: MeSH terms, and V: Venue). For GAHNE [70], we used a learning rate of 0.005, a dropout of 0.5, an L2 penalty of 0.001, and a dimension of 128. For ie-HGCN [132],

Table 5.2: Performance results (AUC score) for the SR task. The best score for each SR is bolded and the second highest is underlined.

Dataset	LINE	GCN	HAN	GAHNE	ie-HGCN	SR-CoMbEr
ACEInhibitors	0.622	0.627	0.649	0.662	<u>0.667</u>	**0.672**
ADHD	0.597	0.605	0.621	0.644	<u>0.646</u>	**0.659**
Antihistamines	0.541	0.544	0.567	<u>0.588</u>	0.586	**0.593**
AtypicalAntipsychotics	0.601	0.607	0.617	<u>0.638</u>	0.636	**0.641**
BetaBlockers	0.629	0.632	0.658	0.671	<u>0.677</u>	**0.684**
CalciumChannelBlockers	0.636	0.64	0.662	<u>0.67</u>	0.666	**0.688**
Estrogens	0.577	0.583	0.607	<u>0.629</u>	0.626	**0.631**
NSAIDs	0.637	0.639	0.662	<u>0.691</u>	0.685	**0.697**
Opioids	0.632	0.635	0.654	0.667	<u>0.671</u>	**0.686**
OralHypoglycemics	0.555	0.559	0.582	<u>0.591</u>	0.583	**0.598**
ProtonPumpInhibitors	0.638	0.641	0.664	0.677	<u>0.681</u>	**0.687**
SkeletalMuscleRelaxants	0.64	0.643	0.658	0.672	<u>0.677</u>	**0.684**
Statins	0.606	0.609	0.633	0.653	<u>0.659</u>	**0.665**
Triptans	0.617	0.624	0.64	0.652	<u>0.66</u>	**0.671**
UrinaryIncontinence	0.633	0.639	0.658	<u>0.678</u>	0.675	**0.683**

the number of layers is set to 5, and the dimension for the four hidden layers starting from the second layer is set to 64, 32, 16, and 8. For SR-CoMbEr, we use $\mathcal{M} = 12$, set the random walk length to 20, and the embedding dimension to 128. The Adam optimizer [58] is used with a learning rate of 0.01 and all parameters are initialized randomly. Dropout is used for all layers except the output layer with a dropout rate of 0.5.

5.3 Empirical Results

The AUC on the Cohen dataset is reported in Table 5.2 for each SR. The best results are bolded and the second-best results are underlined. The results show that HIN embedding significantly outperforms homogeneous network embedding (LINE and GCN). This demonstrates that not only citation information but also other node types (author, venue, and MeSH terms) help to improve the performance of the SR task.

From the table, we observe that SR-CoMbEr outperforms all other baselines from

0.002 to 0.018 by comparing with the second-best AUC score. This indicates the importance of effectively modeling the HIN and demonstrates the effectiveness of SR-CoMbEr in the SR task. Between the existing HIN models, HAN shows the limitation of the user-defined meta-path. The results suggest that there are more hidden but important paths that are difficult for users to define. In contrast, the performance between GAHNE and ie-HGCN is similar. GAHNE performs better when there are more papers excluded from the abstract screening process. For example, the "SkeletalMuscleRelaxants" dataset has a total of 1,643 articles in the beginning but only 34 articles are selected from the abstract screening which is only 2.56%. While GAHNE performs better in cases when fewer articles are selected, ie-HGCN performs better in cases when more papers are selected. For example, "AtypicalAntipsychotics" has a total of 1,120 articles in the beginning and 363 articles passed the screening which is 32.41%.

5.4 Ablation Study

We assess the importance of each component in SR-CoMbEr for the final embedding. *LMV* is a localized, multi-view model that does not use the heterogeneous community detection component (i.e., Section 5.1.1). Each localized, multi-view filter is subsampled using a random walk of all the nodes in the graph. Then the local representations are aggregated using an average function. *CoAvg* extends *LMV* by using the community detection module to construct the localized, multi-view filters. However, unlike the SR-CoMbEr, it does not use the MAE to learn the shared representation from the community filters (i.e., Equation (5.4) is replaced with $H = AVG(H^1, H^2, ..., H^M)$). Table 5.3 summarizes the AUC scores on the test set of the two different multi-view learning techniques on the ACEInhibitors SR task. As shown in the table, incorporating the community information improves the performance (see CoAvg versus LMV).

Table 5.3: Comparison of the AUC score using different community detection algorithms on ACEInhibitors from the SR task.

LMV	CoAvg	CP	SVD	SR-CoMbEr
0.658	0.665	0.668	0.662	**0.672**

By leveraging the community structure, the embedding model can capture different neighborhood information to learn a better representation. While the overall results suggest that although the performance boost is less compared to the community detection component, MAE is beneficial to automatically learn the weights from each of the community representations for the final embedding.

To better understand the importance of the community detection algorithm, we compared the performance using SVD to identify the communities using just one view of the network [99] and CANDECOMP-PARAFAC (CP), a special case of Tucker decomposition where the core tensor only has values along the superdiagonal entries [60]. For SVD, let $F \in \mathbb{R}^{m \times n}$ denote the adjacency matrix of the link type with the largest number of nodes and the target node α. Under SVD, $F = U\Sigma V^*$, where $U \in \mathbb{R}^{m \times p}$, $V \in \mathbb{R}^{n \times p}$ matrix, and $\Sigma \in \mathbb{R}^{p \times p}$. Spectral clustering is then performed in a similar fashion using $\mathcal{M}$ as the number of clusters on the target object, α, and U as the low-dimensional embedding. For CP decomposition, the alternating least square method is used to find the leading left singular values [45]. As shown in the table, SR-CoMbEr (using the HOOI algorithm) for community detection outperforms other techniques (see CP and SVD). While we identify 12 local filters for SR-CoMbEr using HOOI, SVD identifies 9 and CP identifies 14. This shows the importance of identifying the optimal number of filters as using too many or fewer filters can also slightly degrade the performance.

Chapter 6

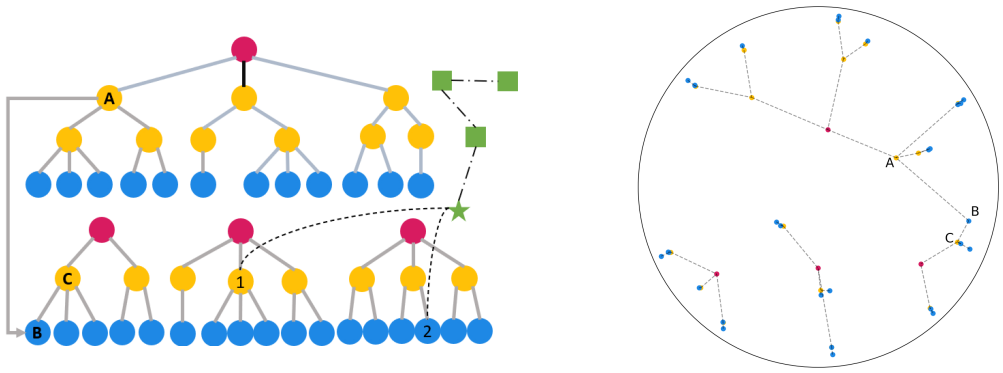
Hyperbolic Representation Learning for Graphs with Mixed Hierarchical and Non-hierarchical Structures

Graphs are popular data structures that contain entities (or nodes) and relations (edges) between nodes. Most real-world graphs are a mixture of hierarchical and non-hierarchical structures. Humans naturally use hierarchy to organize entity categories [94] with typical hierarchical structures denoted as a tree. The structure often consists of an is-a relationship between abstractions such as "Elephant" is-a "Ungulate" and "Ungulate" is-a "Mammal". As a motivating example from PGB, consider the PubMed articles. Articles can cite each other (article-article link) and have a non-hierarchical structure. Each article is also associated with one or more MeSH terms. MeSH terms are controlled and hierarchically organized keywords created and updated regularly by the National Library of Medicine and used for searching biomedical and health-related information. The associated MeSH terms for an article

can be within the same MeSH hierarchy or tree (i.e., terms that are supported by a broader MeSH term) or can be in a different MeSH tree. As a result, knowing the hierarchy relation can play an important role in identifying similar articles as MeSH terms within the same tree are likely to be more similar than those in different trees. Furthermore, it can be difficult to determine if two articles share the same parents without incorporating the MeSH taxonomy. MeSH concepts exhibit a hierarchical structure and consist of multiple trees of different depths. Thus, modeling the PubMed graph with citation and MeSH nodes necessitates handling both hierarchical and non-hierarchical structures.

Unfortunately, most graph representation learning approaches focus on modeling non-hierarchical structures by ignoring the hierarchical nodes or considering the hierarchical (i.e., directed) link as an undirected form. One important characteristic of hierarchical structures is that the number of leaf nodes increases exponentially as the number of levels increases. As such, most graph representation learning approaches suffer from distortion issues when embedding graphs with hierarchical structures. Instead, hyperbolic space has been proposed as an alternative to learning the latent hierarchical structures in the context of embedding models [5, 18, 86, 94, 135]. A key property of hyperbolic space is that the volume grows exponentially with the radius and thus can naturally model the exponential growth in leaf nodes.

Poincaré embedding model [86] is one of the most popular hyperbolic space-based embedding models. The learned node representations are defined within the n-dimensional Poincaré ball such that parallel points along two lines grow exponentially as the points get near the surface of the ball. A Poincaré embedding model implicitly learns the representations of the hierarchy such that root nodes generally lie at the origin while nodes at lower levels of the hierarchy will reside closer to the surface of the ball. Yet there are several limitations to existing Poincaré-based embedding models, as illustrated by our toy example in Figure 6.1. First, they assume



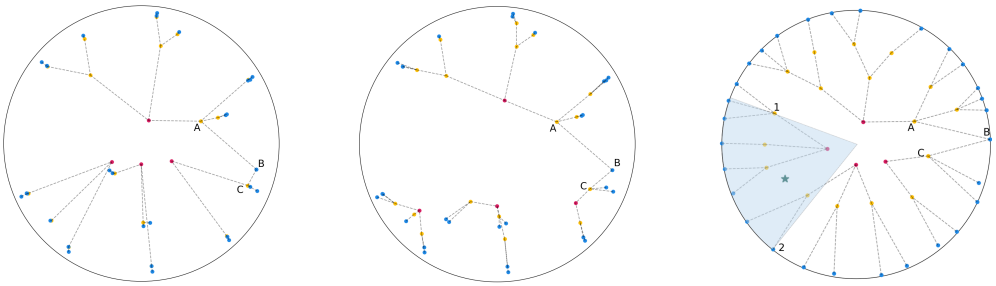
(a) Mixed hierarchical and non-hierarchical structures

(b) Poincaré embedding

Figure 6.1: A toy example of hierarchical structures with four trees, and the results of Poincaré embedding. The circle nodes are from the hierarchical structure, and the star and square nodes are non-hierarchical structures. Some edges are not illustrated in (a) for simplicity. Note that non-hierarchical structures are not shown in (b).

a hierarchical structure with a single root node and may not yield reasonable representations in the presence of multiple nodes (e.g., multiple trees within PubMed). As shown in Figure 6.1(b), none of the 4 roots are embedded close to the origin. Second, when there is a poly-hierarchical structure (i.e., a child can have multiple parents from different trees), the implicit modeling of the hierarchy can result in representations where the child resides closer to the origin than the parent. As shown in Figure 6.1(b), the poly-hierarchical structure results in C and B embedded with a similar distance from the origin. Third, there are only a few hyperbolic embedding models that consider graphs with mixed hierarchical and non-hierarchical structures and leverage the hierarchical structures to learn better representations for the non-hierarchical structures. However, it still remains difficult to transform two different spaces. Last, embedding was developed for the unsupervised setting, recent work has focused on the semi-supervised or supervised setting, partly due to the advantages of graph neural networks.

To address the above limitations, we propose HypMix, an unsupervised **Hyp**erbolic representation learning model for graphs with **Mix**ed hierarchical and non-hierarchical structures. For graphs with hierarchical structures that contain multiple root nodes,



(a) Root Regularization (b) Child Regularization (c) Non-hierarchical structure embedding

Figure 6.2: A toy example of embedding results after applying each component. Note that non-hierarchical structures are not shown in (a) and (b). (c) shows the embedding results of using a hyperbolic entailment cone, and the shadowed area shows the region in the nodes in the non-hierarchical structure can reside. The star node in (c) is the node from a non-hierarchical structure from Figure 6.1(a).

we propose a regularization term to embed the root nodes close to the origin of the Poincaré ball. To tackle challenges related to poly-hierarchical structures, we propose two regularizations: (1) a distance-based restriction to embed parent nodes closer to the origin than their children and (2) using the hyperbolic entailment cone [33] to ensure 2 children reside in a similar Poincaré region. We also introduce the use of the hyperbolic entailment cone to the non-hierarchical structures to better embed these nodes in the Poincaré ball. We conduct extensive experiments across 3 real-world SR tasks to demonstrate the effectiveness of HypMix over existing baselines. We also perform an ablation study to better understand the benefits of the three components of our model.

6.1 HypMix

HypMix adopts Poincaré embedding [86] which learns the representation of hierarchical structure into a hyperbolic space or an n-dimensional Poincaré ball. However, the basic Poincaré embedding model does not always learn the representation that preserves the hierarchical structure. For example, Poincaré embedding model cannot

handle multiple root nodes which leads the root nodes to be placed in the outer part of the hyperbolic space than their child nodes as shown in Figure 6.1(a). Also, some parent nodes are located further from the origin compared to their child node because of the poly-hierarchical structures. Another limitation of Poincaré embedding is that it is a model only for hierarchical structure which makes it difficult to learn the representation with non-hierarchical structures. To resolve these limitations, we use two regularizations to learn a better representation of the hierarchy structure and use hyperbolic entailment cone [33] to also learn the representation of non-hierarchical structures.

6.1.1 Root Regularization

One limitation of existing Poincaré-based models is the implicit design for a hierarchical structure with a limited number of roots (i.e., a small number of trees). However, some hierarchical taxonomies may have multiple categories or concepts that can be further separated into subcategories. For example, the MeSH contains 115 root nodes. Unfortunately, when the hierarchical structure encompasses multiple trees, the root embeddings of the tree may reside closer to the surface of the Poincaré ball (as shown in Figure 6.1). This restricts the embedding space to learn the hierarchical structure of subsequent children nodes and thus may result in suboptimal leaf embeddings.

To address the limitation of existing Poincaré-based embedding models for hierarchical structures with multiple root nodes, we propose a regularization term to encourage the root node to reside close to the origin. In this manner, the subtree has sufficient space and more flexibility to better preserve deeper trees. Formally, given a root node, $root$, we denote the distance to the origin, $origin$ as $d(origin, root)$ and require the root node to be within a certain δ such that $d(origin, root) < \delta$. The distance metric is denoted in Equation 2.4. Figure 6.2(a) shows the results of root regularization with our motivating toy example shown in Figure 6.1. As shown in

Figure 6.1(b), an unconstrained Poincaré-based embedding model may place one or more of the root nodes near the surface of the hyperbolic space. In comparison, our root regularization term, shown in Figure 6.2(a), will force all the root nodes to lie near the origin.

6.1.2 Child Regularizations

Another limitation of the Poincaré embedding model is that it only implicitly captures the hierarchical structure. As such, it may not be able to distinguish which node is a child or parent and place child nodes closer to the origin than their parents. This is particularly difficult for a poly-hierarchical structure where a node may have parents from different trees. For example, MeSH is a poly-hierarchical ontology where concepts can belong to multiple categories. Figure 6.1(b) illustrates an example of where the Poincaré confuses a child and parent due to the poly-hierarchical structure. In this scenario, the ideal representation is the parent embedding residing closer to the origin than the child.

Distance-based Child Regularization.

We first propose a regularization term that restricts a parent from being further in distance from the origin than its child. Similar to the root regularization, given a *parent* node and a *child* node, we compute 2 additional distances, $d(origin, parent)$ and $d(origin, child$ using Equation (2.4). We then enforce the model to learn a representation such that $d(origin, parent) < d(origin, child)$. Figure 6.2(b) demonstrates the learned embedding after the child regularization is applied to Figure 6.1(b). We briefly note that root regularization is not applied in this scenario. As shown in the figure, the child node resides further from the origin than its parent node and explicitly preserves the hierarchical structure where nodes at lower levels will be closer to the surface of the ball.

Hyperbolic Entailment Cone Regularization.

The above distance-based child regularization can help preserve the relationship between one parent and one child, yet may not ensure two children of the same parent reside in a "similar" Poincaré region. As such, we posit that a partial ordering where each subtree naturally defines the Poincaré region can further improve the learned embedding of the nodes within the tree. The idea is that a parent node will define a cone in the Poincaré space for which its children can be placed and allows better differentiation of the node embeddings between multiple trees. Thus if a child shares two parents, then it can only be nested in the intersection of the two cones defined by the parents. To achieve this, we leverage the hyperbolic entailment cone [33] to place the children nodes within the hyperbolic cones defined by the parent. Figure 6.1(c) shows the illustration of using a hyperbolic entailment cone in a hierarchical structure.

6.1.3 Non-hierarchical Structure Embedding

Across many real-world graphs, their nodes may capture both hierarchical and non-hierarchical structures. The above regularizations (root, distance-based root, and hyperbolic entailment cone) can preserve the hierarchical structures, yet do not account for links to nodes that may not have a non-hierarchical structure. As a motivating example, each PubMed article can be tagged with multiple MeSH categories (which exhibit a hierarchical structure) yet the articles themselves do not have a hierarchical structure. As such, the natural question is how to leverage the hierarchical structure to better embed the non-hierarchical nodes in the hyperbolic space.

Suppose we have two node types, $H = \{h_1, h_2, ..., h_n\}$ and $V = \{v_1, v_2, ..., v_m\}$, where there is a hierarchical relationship between the nodes in H, the nodes in V have non-hierarchical structure (i.e., can be linked to each other but not as a parent-child relationship), and there are non-hierarchical links between H and V. Note that any

nodes in V can have multiple relations with the nodes in H, and linked nodes in H can be located at any level in the tree. In this scenario, the hierarchical structure of H can serve as a guideline to learn the representation of the nodes in V. Our idea is that any node v_i that is linked to a node in h_k should then naturally reside in the same angular cone region defined by the node through the hyperbolic entailment cone. Thus, nodes in a non-hierarchical structure are also subject to residing within the hyperbolic entailment cone of the hierarchical nodes. The blue area in Figure 6.1(c) is the region where the nodes in non-hierarchical structures can be located.

6.2 Experimental Design

6.2.1 Data Preprocessing

We evaluate our model on 15 SRs provided by Cohen *et al.* [26], 3 SRs provided by SWIFT-Review [49], and 3 SRs from CLEF-TAR [55] dataset. We follow the experimental setting used for PGB [65]. The detailed information is explained in the Section 4.4.1. Based on each SR dataset, a subset of the entire PGB graph is constructed and the graph is split into three train-test trials of 70%-30%, respectively. Table 2.1 summarizes the statistics for the abstract screening process for each SR topic across the 3 SR datasets. For the SR task, we use two node types, Paper (P) and MeSH terms (M), and three edge types, P – P, P – M, and M – M. The edge type, M – M, has a hierarchical structure and others are non-hierarchical.

6.2.2 Statistics of Hierarchical Structures

Hyperbolicity is a measure of how hierarchical the graph is and is defined as follows [39]:

Definition 2. Hyperbolicity. Let a,b,c,d are the nodes of the graph, let S_1, S_2,

Table 6.1: Statistics of the MeSH hierarchies.

Statistics	Value
# of root nodes	115
# of total nodes	30191
Avg. of # of leaf nodes	391.93
Avg. of # of nodes	541.89
Avg. of Depth	6.43
Avg. of Hyperbolicity	3.53

and S_3 be defined by

$$\begin{aligned} S_1 &= dist(a,b) + dist(d,c) \\ S_2 &= dist(a,c) + dist(b,d) \\ S_3 &= dist(a,d) + dist(b,c) \end{aligned} \tag{6.1}$$

and let M_1 and M_2 be the two largest values between S_1, S_2, and S_3. We define $hyp(a,b,c,d) = M_1 - M_2$, and the hyperbolicity $\delta(G)$ of the graph is the maximum of hyp over all possible 4-tuples (a,b,c,d) divided by 2. That is, the graph is said δ-hyperbolic when

$$\delta(G) = \frac{1}{2} max_{a,b,c,d \in V(G)} hyp(a,b,c,d) \tag{6.2}$$

A hyperbolicity value of 0 means the graph is a tree. In other words, a hyperbolicity value closer to 0 means that the graph is more hierarchical. The hyperbolicity of a graph is the maximum over all its biconnected components.

Table 6.1 provides summary statistics of the hierarchical structures of the MeSH tree. MeSH trees have 115 unique trees with a total of 30,191 unique MeSH terms. The results suggest that most of the individual trees are wide (the average number of leaf nodes is 391.93) but not deep (average depth is 6.43). The MeSH hierarchy structure also contains the poly-hierarchical structure where one node has multiple parents from two different trees. For example, there are two roots, *Body Regions* and *Sense Organs*. Both trees contains *Eye* as a child node such as $BodyRegions \rightarrow Head \rightarrow Face \rightarrow Eye$, and $SenseOrgans \rightarrow Eye$. As shown in the table, the average

hyperbolicity of each tree is 3.53 which suggests that MeSH terms might benefit from explicitly modeling this hierarchical structure.

6.2.3 Baseline Models

In this section, we discuss the baseline model that we use to evaluate our model. We also analyze each component that we propose. As HypMix is an unsupervised model, we also compare it with a conventional network embedding model that uses the Euclidean space. We use a softmax layer to learn the classifier. For LINE, we use the dimension size $d = \{50, 100, 256\}$ which means that for $d = 50$, we use 25 for each first- and second-proximity, and for $d = 100$, we use 50 for each proximity. For other hyperbolic embedding models, we use $d = \{2, 10, 20, 30, 50\}$ for the dimension size. For all the baselines, we used a single g4dn AWS instance with NVIDIA T4 GPU.

- LINE [107]: LINE is a conventional network embedding model that uses first- and second-proximity using the joint probability between two nodes. LINE is an unsupervised network embedding model which learns the representation in the Euclidean space.

- Poincaré Embedding [86]: Poincaré Embedding model learns the representation in the hyperbolic space. This is the model that does not use any regularization or hyperbolic entailment cones for learning the representations.

- HypMix$_{root}$: On top of Poincaré embedding model, this setting only applies the root regularization technique.

- HypMix$_{child}$: On top of Poincaré embedding model, this setting only applies the child regularization technique.

- HypMix$_{cone}$: In addition to all the regularization techniques (root and child regularization), this setting uses the hyperbolic entailment cone to embed the

hierarchical structure but does not use it for non-hierarchical structures.

- HypMix: This setting uses all the techniques we proposed.

6.3 Empirical Results

The AUC score on the three splits is reported in Table 6.2 for each SR. The results are on average of the AUC score of three trials, and the reported results use using dimension size of 50 for all the baseline models. The best results are bolded and the second-best results are underlined. The results show that HypMix outperforms the other baseline models (LINE and Poincaré). This demonstrates that not only citation information but also MeSH hierarchy information helps to improve the performance of the SR task.

From the table, we observe that HypMix outperforms all other baselines from 0.005 to 0.024 by comparing with the second-best AUC score. This indicates the importance of effectively modeling both the hierarchical and non-hierarchical structures. Moreover, it demonstrates the effectiveness of HypMix in the SR task. Between the original Poincaré embedding model and HypMix, the results show that HypMix significantly outperforms the former and highlights the effectiveness of the components that we propose. It also shows that the original Poincaré embedding model cannot handle multiple trees and mixed node types. By comparing the results with LINE which uses Euclidean space, HypMix outperforms LINE which illustrates the the importance of using hyperbolic space for hierarchical relations. Although LINE uses a dimension size of 256, HypMix outperforms LINE by only using the dimension size 50. In addition, in some cases, HypMix$_{cone}$ outperforms LINE.

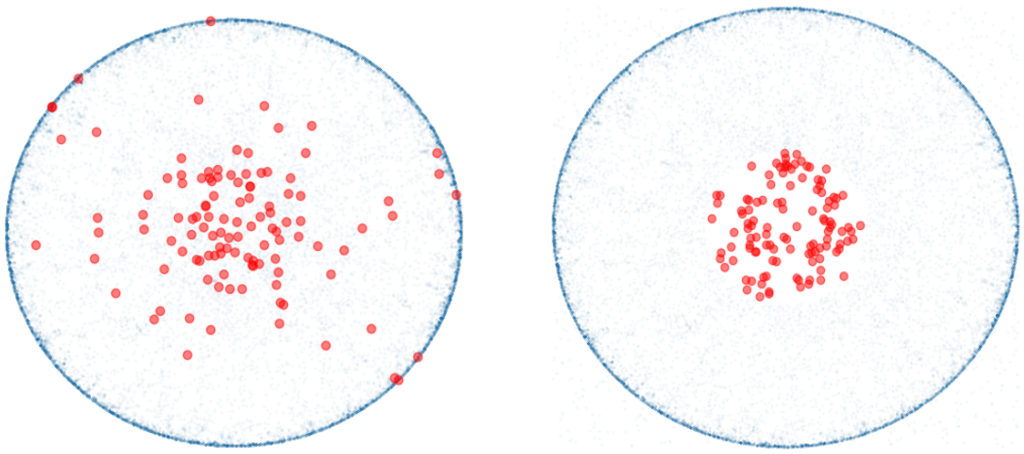
Although each of the regularization techniques can improve performance, it is not consistent across all the 21 SR reviews. For the root regularization, HypMix$_{root}$, there are slightly more mixed results as the improvement ranges between -0.005 to

Table 6.2: Performance results (AUC score) for the SR task. The best score for each SR is bolded and the second highest is underlined.

Dataset	LINE	Poincaré	HypMix$_{root}$	HypMix$_{child}$	HypMix$_{cone}$	HypMix
Cohen-ACEInhibitors	<u>0.572</u>	0.524	0.534	0.532	0.556	**0.589**
Cohen-ADHD	0.538	0.522	0.523	0.533	<u>0.539</u>	**0.552**
Cohen-Antihistamines	<u>0.552</u>	0.518	0.514	0.534	0.547	**0.567**
Cohen-AtypicalAntipsychotics	<u>0.556</u>	0.522	0.523	0.534	0.552	**0.561**
Cohen-BetaBlockers	<u>0.581</u>	0.554	0.551	0.555	0.579	**0.59**
Cohen-CalciumChannelBlockers	<u>0.588</u>	0.549	0.555	0.559	0.581	**0.599**
Cohen-Estrogens	<u>0.542</u>	0.53	0.529	0.534	0.539	**0.548**
Cohen-NSAIDS	0.564	0.536	0.535	0.54	<u>0.568</u>	**0.588**
Cohen-Opioids	<u>0.592</u>	0.544	0.539	0.546	0.583	**0.606**
Cohen-OralHypoglycemics	<u>0.511</u>	0.502	0.502	0.504	0.51	**0.535**
Cohen-ProtonPumpInhibitors	<u>0.592</u>	0.523	0.527	0.533	0.585	**0.61**
Cohen-SkeletalMuscleRelaxants	<u>0.594</u>	0.534	0.532	0.542	0.581	**0.612**
Cohen-Statins	0.556	0.534	0.543	0.542	<u>0.558</u>	**0.577**
Cohen-Triptans	<u>0.573</u>	0.53	0.534	0.544	0.565	**0.596**
Cohen-UrinaryIncontinence	<u>0.597</u>	0.537	0.542	0.543	0.569	**0.609**
SWIFT-Transgenerational	0.628	0.566	0.579	0.577	<u>0.632</u>	**0.645**
SWIFT-PFOS-PFOA	<u>0.629</u>	0.572	0.581	0.573	0.622	**0.641**
SWIFT-BPA	<u>0.558</u>	0.518	0.524	0.523	0.552	**0.57**
CLEF-Prognosis-CD012661	0.573	0.532	0.54	0.538	<u>0.576</u>	**0.598**
CLEF-DTA-CD008803	<u>0.582</u>	0.544	0.554	0.552	0.579	**0.604**
CLEF-Intervention-CD005139	<u>0.603</u>	0.556	0.566	0.561	0.596	**0.627**

0.013. For the child regularization, HypMix$_{child}$, the improvement of the score is 0.001 to 0.016. This indicates that the root regularization itself does not always help to learn the better representation as there are more nodes that are not root. Once we apply the hyperbolic entailment cone, the performance significantly improves. By comparing with the Poincaré embedding model, HypMix$_{cone}$ outperforms from 0.008 to 0.066. This shows that the hyperbolic entailment cone plays an important role in learning a representation of hierarchical relations.

Poincaré embedding, HypMix$_{root}$, and HypMix$_{child}$ have similar results. The difference between Poincaré embedding and the other two settings is relatively small from -0.002 to 0.001. This shows that the original Poincaré embedding works with a small hierarchical structure. Once the hyperbolic entailment cone is applied, the performance significantly increases which is similar to the SR task. For both datasets, HypMix$_{cone}$ even outperforms LINE by 0.026 for DBLP, and 0.028 for the YELP dataset. The table shows that HypMix significantly outperforms all the baseline mod-



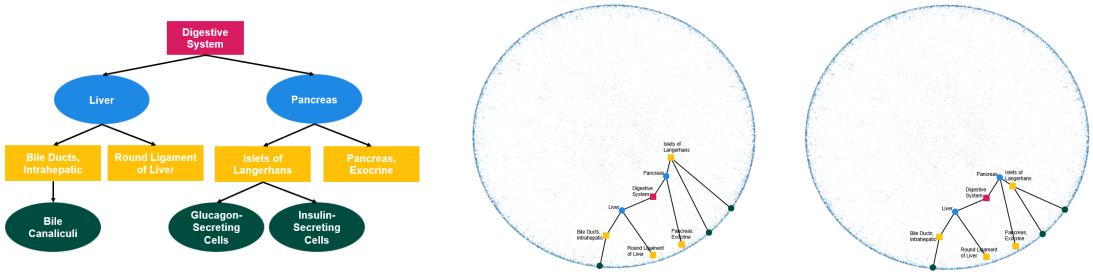
(a) Poincaré Embedding (b) After root regularization

Figure 6.3: Examples of a Poincaré Embedding with the dimension size of two using the MeSH hierarchy. The red dot denotes the root nodes and the blue dots denote other nodes.

els. HypMix outperforms LINE with 0.051 and 0.041 in the AUC score for DBLP and YELP, respectively. Also HypMix outperforms HypMix$_{cone}$ with 0.025 for DBLP and 0.013 for YELP. This means that the hyperbolic entailment cone regularization is important to learn a better representation of hierarchical structure and also it well-supports to embed non-hierarchical structures.

6.4 Case Study

One limitation of Poincaré embedding is the difficulty of handling multiple trees. To better understand this, we perform a case study on the MeSH hierarchy. Figure 6.3 depicts an example of the original Poincaré Embedding using the MeSH hierarchy where the red dots are the root nodes and all other nodes are blue dots. As can be seen, the roots are distributed throughout the hyperbolic space. This limits the space to learn the representation of the hierarchical structure. For example, if the root resides closer to the surface of the ball then there will be not enough space to embed the children of the root. Once we apply our root regularization, the roots will be located close to the origin of the space, as shown in Figure 6.3(b). This shows



(a) Partial Mesh Hierarchy

(b) Poincaré Embedding

(c) After child regularization

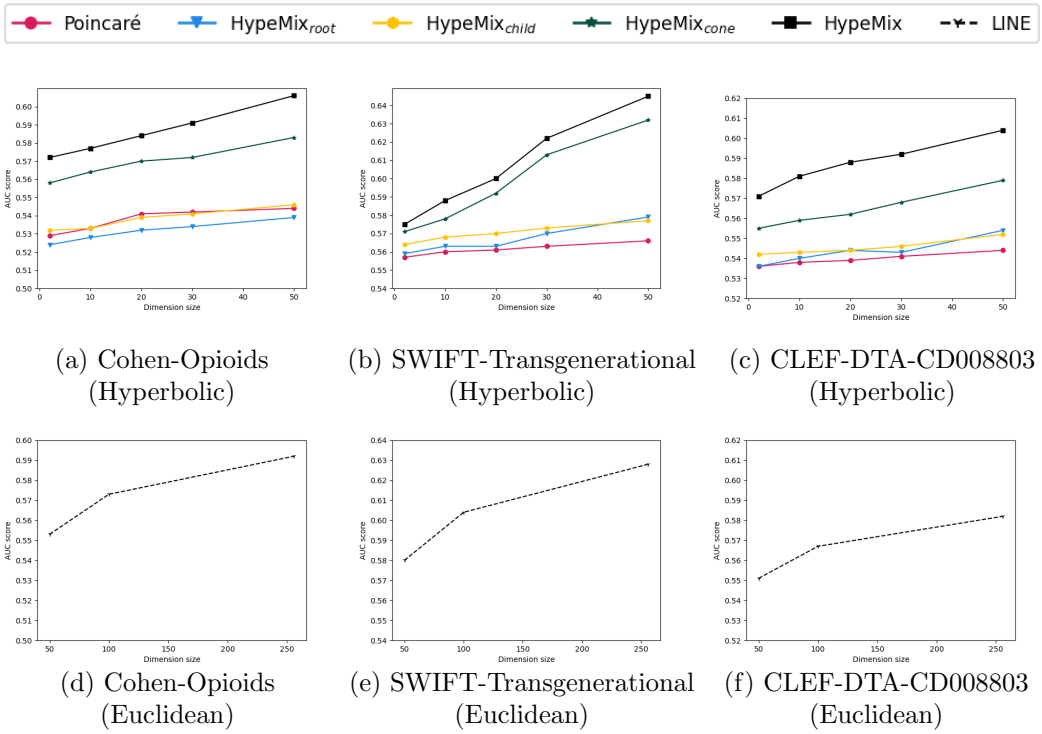
Figure 6.4: Examples of a Poincaré Embedding with the dimension size of two using the MeSH hierarchy. (a) shows the partial MeSH hierarchy that is used to illustrate the embedding results in (b) and (c).

that root regularization helps to reserve enough space to capture the subtrees.

Another limitation of Poincaré embedding is that it implicitly learns the hierarchical structure. As a result, the child nodes can reside closer to the origin compared to the parent node as shown in Figure 6.4. Figure 6.4(a) shows the partial MeSH tree for the highlighted points in Figures 6.4(b) and (c). Note that only child regularization is applied in Figure 6.4(c). As shown in the figure, *Islets of Langerhands* is a child node of *Pancreas*. However, in Figure 6.4(b) for the Poincaré embedding, *Islets of Langerhands* is located closer to the origin than *Pancreas*. Once we apply child regularization, as shown in 6.4(c), *Pancreas* is embedded closer to the origin than *Islets of Langerhands*. However, still *Pancreas* is located closer to the origin compared to its parent node *Digestive System* and suggests not only child regularization is important, but also root regularization is necessary to reserve enough space for the descendent nodes.

6.5 Impact of Dimension Size

We also compare the model performance with different dimension sizes for hyperbolic and Euclidean space. For LINE which uses the Euclidean space for graph representation learning, we use the dimension size $d = \{50, 100, 256\}$. For the hyperbolic



(a) Cohen-Opioids (Hyperbolic)

(b) SWIFT-Transgenerational (Hyperbolic)

(c) CLEF-DTA-CD008803 (Hyperbolic)

(d) Cohen-Opioids (Euclidean)

(e) SWIFT-Transgenerational (Euclidean)

(f) CLEF-DTA-CD008803 (Euclidean)

Figure 6.5: Comparison of the performance with different dimension sizes on Hyperbolic and Euclidean space.

embeddings, we use $d = \{2, 10, 20, 30, 50\}$ for the dimension size. Figure 6.5 shows the results from three SR tasks, Cohen-Opioids, SWIFT-Transgenerational, and CLEF-DTA-CD00803. The hyperbolic results show a similar trend where Poincaré embedding, HypMix$_{root}$, and HypMix$_{child}$ have similar results as the dimension size increases. The performance of HypMix$_{cone}$ and HypMix increases as the dimension size increases. These results show that to learn the representation of a complex hierarchical structure, the model requires a higher dimension size. For the Euclidean space, we see a similar trend as HypMix$_{cone}$ and HypMix, however, it is important that the dimension size of the hyperbolic space is still smaller than the Euclidean space and offers a better performance.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, we propose several models for automating the abstract screening process for SR using network embedding models.

We first proposed a model which uses the Multi-modal Missing Data aware Stacked Autoencoder (MMiDaS-AE) — inspired by [15] — for biomedical citation screening. We showed that this multi-modal approach, which treats title/abstract texts, citation networks, and topics as separate modalities and explicitly models these, outperforms prior models in inter-topic settings. Further, in the topic-specific (intra-topic) setting, our fine-tuned MMiDaS-AE outperforms alternative approaches.

Second, we illustrate the rich metadata fields found in biomedical literature. Although there exist many studies that use HIN embedding for various tasks such as node classification, link prediction, and SR, no existing data fully captures all the available information found in PubMed. We construct PGB, a biomedical literature bibliographic dataset, that contains 11 fields of metadata. The strength of PGB is not only that it contains multiple types of nodes and edges, but also captures a hierarchical structure on one of its nodes.

Third, we propose SR-CoMbEr to learn citation network representations for SRs using the rich metadata in PGB. To avoid defining the meta-path, we formulate the problem using multi-view learning to automatically capture the semantics of HIN. To encode the structural heterogeneity and neighborhood information, we use community detection and multiple community-based views of the network and fuse the representations to obtain the final representation. We also introduce the use of HOOI to compute the optimal number of filters in concert with community detection. The experiments on 15 SR topics show that SR-CoMbEr outperforms several state-of-the-art HIN embedding models.

Finally, we propose HypMix, an unsupervised hyperbolic representation learning for graphs with mixed hierarchical and non-hierarchical structures, to better capture the hierarchical structure of the MeSH terms. We address the limitations of the Poincaré embedding model for handling multiple roots and poly-hierarchical structures. We propose root regularization to learn the representations of the root nodes to locate closer to the origin of the hyperbolic space. We propose child regularizations so that the parent node will reside closer to the origin than its child nodes, and the child will reside in the region of the parent. Also to learn the representation of the non-hierarchical structure, we adopt a hyperbolic entailment cone by defining the region for the hierarchical structures so that we can also define the region of the non-hierarchical nodes. The extensive experiments on 21 real-world SR tasks show that HypMix outperforms existing unsupervised graph representation learning models. The case study shows the importance of each component and also shows that hyperbolic space performs better than Euclidean space with a smaller dimension size.

7.2 Future Work

This dissertation can be extended from the following aspects.

- Add supervision to HypMix: The model we proposed for the hierarchical embedding is an unsupervised method. A supervised model for hierarchical embedding has not been extensively studied, however, supervision should play an important role in our task. Some work [104, 133] were proposed to use supervision in hierarchical embedding, but are not exactly related to our task as they do not tackle the mixed structure.

- Combination of HIN embedding and hierarchical embedding: We showed the effectiveness of incorporating the metadata available in biomedical literature. We provided PGB, a new PubMed benchmark, which can help to extract metadata information easily. We proposed MMiDaS-AE to illustrate the importance of citation network, SR-CoMbEr to demonstrate how incorporating non-hierarchical metadata improves the result by using HIN embedding, and HypMix to display the effectiveness of incorporating hierarchical information. The next step can entail combining HIN embedding and hierarchical embedding, however, the difficulty is combining Euclidean space (HIN embedding) and hyperbolic space (hierarchical embedding).

- Adding text information: Most of the existing works [46, 91, 117, 119] are simple text-based methods such as bag-of-words or TF-IDF. This shows that the text information (title and abstract) is easy to access and also important in SR tasks. Thus, on top of HIN embedding including the hierarchical structures, using language models should play an important role that can significantly improve the performance of SR.

- Exploration on Large Language Model: Recent studies proposed powerful large language models such as LLaMA [109] and generative AI models such as ChatGPT [90]. However, there are only a few works [126] that use a large language model on SR, and none of them compare the performance with the network em-

bedding model using the metadata. It is important to understand the impact of the large language model, and it is necessary to compare the performance with the network embedding using the metadata. Also, we can serialize the metadata with the textual data into a large language model to improve the performance of SR.

Bibliography

[1] JJ García Adeva, JM Pikatza Atxa, M Ubeda Carrillo, and E Ansuategi Zengotitabengoa. Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, 41(4):1498–1508, 2014.

[2] I Elaine Allen and Ingram Olkin. Estimating time to conduct a meta-analysis from number of citations retrieved. *JAMA*, 282(7):634–635, August 1999.

[3] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *Proc. of NAACL-HLT*, 2018.

[4] Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. Taxocomplete: Self-supervised taxonomy completion leveraging position-enhanced semantic matching. In *Proceedings of the ACM Web Conference 2023*, pages 2509–2518, 2023.

[5] Yushi Bai, Zhitao Ying, Hongyu Ren, and Jure Leskovec. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. *Advances in Neural Information Processing Systems*, 34:12316–12327, 2021.

[6] Ethan Balk, Gowri Raman, Mei Chung, Stanley Ip, Athina Tatsioni, Alvaro Alonso, Priscilla Chew, Scott J Gilbert, and Joseph Lau. Effectiveness of management strategies for renal artery stenosis: a systematic review. *Annals of internal medicine*, 145(12):901–912, 2006.

[7] Alexandra Bannach-Brown, Piotr Przybyła, James Thomas, Andrew SC Rice, Sophia Ananiadou, Jing Liao, and Malcolm Robert Macleod. Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error. *Systematic reviews*, 8(1): 1–12, 2019.

[8] Hilda Bastian, Paul Glasziou, and Iain Chalmers. Seventy-five trials and eleven systematic reviews a day: How will we ever keep up? *PLOS Medicine*, 7(9): e1000326, September 2010.

[9] Tanja Bekhuis and Dina Demner-Fushman. Towards automating the initial screening phase of a systematic review. In *MedInfo*, pages 146–150, 2010.

[10] Iz Beltagy, Arman Cohan, and Kyle Lo. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[11] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[12] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[13] Rohit Borah, Andrew W Brown, Patrice L Capers, and Kathryn A Kaiser. Analysis of the time and workers needed to conduct systematic reviews of medical

interventions using data from the prospero registry. *BMJ open*, 7(2):e012545, 2017.

[14] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[15] Cesar Cadena, Anthony R Dick, and Ian D Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, volume 5, page 1, 2016.

[16] Kathi Canese and Sarah Weis. Pubmed: the bibliographic database. *The NCBI Handbook*, 2013.

[17] Iain Chalmers, Larry V Hedges, and Harris Cooper. A brief history of research synthesis. *Evaluation & the Health Professions*, 25(1):12–37, June 2016.

[18] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[19] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2005.00545*, 2020.

[20] Jackie Chandler, Rachel Churchill, Julian Higgins, Toby Lasserson, David Tovey, et al. Methodological standards for the conduct of new cochrane intervention reviews. *Sl: Cochrane Collaboration*, 2013.

[21] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[22] Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 94–102, 2022.

[23] Mei Chung, Ethan M Balk, Stanley Ip, Gowri Raman, Winifred W Yu, Thomas A Trikalinos, Alice H Lichtenstein, Elizabeth A Yetley, and Joseph Lau. Reporting of systematic reviews of micronutrients and health: a critical appraisal. *The American journal of clinical nutrition*, 89(4):1099–1113, 2009.

[24] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*, 2020.

[25] Aaron M Cohen. Optimizing feature representation for automated systematic review work prioritization. In *AMIA annual symposium proceedings*. American Medical Informatics Association, 2008.

[26] Aaron M Cohen, William R Hersh, Kim Peterson, and Po-Yin Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, 13(2):206–219, 2006.

[27] Aaron M Cohen, Kyle Ambert, and Marian McDonagh. Cross-topic learning for work prioritization in systematic review creation and update. *Journal of the American Medical Informatics Association*, 16(5):690–704, 2009.

[28] L De, B De-Moor, and J Vandewalle. On the best rank-1 and rank-(r1 r2... rn) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4): 1324–1342, 2000.

[29] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[31] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[32] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1797–1806, 2017.

[33] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pages 1646–1655. PMLR, 2018.

[34] Zelalem Gero and Joyce Ho. Pmcvec: Distributed phrase representation for biomedical text processing. *Journal of Biomedical Informatics: X*, 3:100047, 2019.

[35] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12): 7821–7826, 2002.

[36] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learn-

ing in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* IEEE, 2005.

[37] David Gough and Diana Elbourne. Systematic research synthesis to inform policy, practice and democratic debate. *Social Policy and Society*, 1(3):225–236, July 2002.

[38] David Gough, Sandy Oliver, and James Thomas. *An introduction to systematic reviews.* Sage, second edition, 2017.

[39] Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer, 1987.

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

[41] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International conference on learning representations*, 2018.

[42] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[43] Neal R Haddaway and Martin J Westgate. Predicting the time needed for environmental systematic reviews and systematic maps. *Conservation Biology*, 6:136, October 2018.

[44] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

[45] Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[46] Kazuma Hashimoto, Georgios Kontonatsios, Makoto Miwa, and Sophia Ananiadou. Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of biomedical informatics*, 62:59–65, 2016.

[47] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[48] Brian E Howard, Jason Phillips, Kyle Miller, Arpit Tandon, Deepak Mav, Mihir R Shah, Stephanie Holmgren, Katherine E Pelch, Vickie Walker, Andrew A Rooney, Malcolm Macleod, Ruchir R Shah, and Kristina Thayer. Swift-review: A text-mining workbench for systematic review. *Systematic Reviews*, 5(1):87, May 2016.

[49] Brian E Howard, Jason Phillips, Kyle Miller, Arpit Tandon, Deepak Mav, Mihir R Shah, Stephanie Holmgren, Katherine E Pelch, Vickie Walker, Andrew A Rooney, et al. Swift-review: a text-mining workbench for systematic review. *Systematic reviews*, 5:1–16, 2016.

[50] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[51] Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J. Marshall, and Byron C. Wallace. Learning disentangled representations of texts with application to biomedical abstracts. In *Proc. of EMNLP*, 2018.

[52] Hyunchul Jang, Jaesoo Lim, Joon-Ho Lim, Soo-Jun Park, Kyu-Chul Lee, and Seon-Hee Park. Finding the evidence for protein-protein interactions from PubMed abstracts. *ArXiv preprint*, 2022.

[53] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[54] Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. Taxoenrich: Self-supervised taxonomy completion via structure-semantic representations. In *Proceedings of the ACM Web Conference 2022*, pages 925–934, 2022.

[55] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. Clef 2019 technology assisted reviews in empirical medicine overview. In *CEUR workshop proceedings*, volume 2380, 2019.

[56] Tom Kenter, Alexey Borisov, and Maarten De Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*, 2016.

[57] Madian Khabsa, Ahmed Elmagarmid, Ihab Ilyas, Hossam Hammady, and Mourad Ouzzani. Learning to identify relevant studies for systematic reviews using random forest and external information. *Machine Learning*, 102:465–482, 2016.

[58] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[59] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*, 2017.

[60] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[61] Georgios Kontonatsios, Austin J Brockmeier, Piotr Przybyła, John McNaught, Tingting Mu, John Y Goulermas, and Sophia Ananiadou. A semi-supervised approach using label propagation to support citation screening. *Journal of biomedical informatics*, 72:67–76, 2017.

[62] Ioannis Koulouridis, Mansour Alfayez, Thomas A Trikalinos, Ethan M Balk, and Bertrand L Jaber. Dose of erythropoiesis-stimulating agents and adverse outcomes in ckd: a metaregression analysis. *American Journal of Kidney Diseases*, 61(1):44–56, 2013.

[63] Martin Krallinger, Florian Leitner, and Alfonso Valencia. Analysis of biological processes and diseases using text mining approaches. *Methods in molecular biology*, 2010.

[64] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[65] Eric W Lee and Joyce C Ho. Pgb: A pubmed graph benchmark for heterogeneous network representation learning. *arXiv preprint arXiv:2305.02691*, 2023.

[66] Eric W Lee and Joyce C Ho. Sr-comber: Heterogeneous network embedding using community multi-view enhanced graph convolutional network for automating systematic reviews. In *European Conference on Information Retrieval*, pages 553–568. Springer, 2023.

[67] Eric W Lee, Byron C Wallace, Karla I Galaviz, and Joyce C Ho. Mmidas-ae: multi-modal missing data aware stacked autoencoder for biomedical abstract screening. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020.

[68] Tao Lei, Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Deriving neural architectures from sequence and graph kernels. In *International Conference on Machine Learning*, pages 2024–2033. PMLR, 2017.

[69] Ivan Lerner, Perrine Créquit, Philippe Ravaud, and Ignacio Atal. Automatic screening using word embeddings achieved high sensitivity and workload reduction for updating living network meta-analyses. *Journal of clinical epidemiology*, 108:86–94, 2019.

[70] Xiaohe Li, Lijie Wen, Chen Qian, and Jianmin Wang. Gahne: Graph-aggregated heterogeneous network embedding. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020.

[71] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *Proc. of ICLR*, 2016.

[72] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proc. of ACL*, 2020.

[73] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

[74] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.

[75] Stan Matwin, Alexandre Kouznetsov, Diana Inkpen, Oana Frunza, and Peter O'Blenis. A new algorithm for reducing the workload of experts in performing

systematic reviews. *Journal of the American Medical Informatics Association*, 17(4):446–453, 2010.

[76] Manuele Michelessi, Ersilia Lucenteforte, Francesco Oddone, Miriam Brazzelli, Mariacristina Parravano, Sara Franchi, Sueko M Ng, and Gianni Virgili. Optic nerve head and fibre layer imaging for diagnosing glaucoma. *Cochrane Database of Systematic Reviews*, (11), 2015.

[77] Makoto Miwa, James Thomas, Alison O'Mara-Eves, and Sophia Ananiadou. Reducing systematic review workload through certainty-based screening. *Journal of biomedical informatics*, 51:242–253, 2014.

[78] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, and The PRISMA Group. Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *PLOS Medicine*, 6(7):e1000097, July 2009.

[79] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[80] Zoë Slote Morris, Steven Wooding, and Jonathan Grant. The answer is 17 years, what is the question: understanding time lags in translational research. *Journal of the Royal Society of Medicine*, 104(12):510–520, December 2011.

[81] Sally Morton, Alfred Berg, Laura Levit, Jill Eden, et al. *Finding what works in health care: standards for systematic reviews*. National Academies Press, 2011.

[82] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and UMD EDU. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, 2012.

[83] Mark EJ Newman. Detecting community structure in networks. *The European physical journal B*, 38(2):321–330, 2004.

[84] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[85] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[86] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6341–6350. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7213-poincare-embeddings-for-learning-hierarchical-representations.pdf`.

[87] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4920–4928, 2016.

[88] Christopher Norman, Mariska Leeflang, Pierre Zweigenbaum, and Aurélie Névéol. Automating document discovery in the systematic review process: How to use chaff to extract wheat. In *International Conference on Language Resources and Evaluation*, 2018.

[89] Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, 4(1):5, 2015.

[90] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

[91] Mourad Ouzzani, Hossam Hammady, Zbys Fedorowicz, and Ahmed Elmagarmid. Rayyan-a web and mobile app for systematic reviews. *Systematic reviews*, 5(1):210, 2016.

[92] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8 (2):1–44, 2016.

[93] Pascal Vincent PASCALVINCENT and Hugo Larochelle LAROCHEH. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion pierre-antoine manzagol. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[94] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

[95] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[96] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[97] Bernd Richter, Bianca Hemmingsen, Maria-Inti Metzendorf, and Yemisi Takwoingi. Development of type 2 diabetes mellitus in people with intermediate hyperglycaemia. *Cochrane Database of Systematic Reviews*, (10), 2018.

[98] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.

[99] Somwrita Sarkar and Andy Dong. Community detection in graphs using singular value decomposition. *Physical Review E*, 83(4):046114, 2011.

[100] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[101] Harrisen Scells, Guido Zuccon, Bevan Koopman, Anthony Deacon, Leif Azzopardi, and Shlomo Geva. A test collection for evaluating retrieval of studies for inclusion in systematic reviews. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1237–1240, 2017.

[102] Jingbo Shang, Meng Qu, Jialu Liu, Lance M Kaplan, Jiawei Han, and Jian Peng. Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. *arXiv preprint arXiv:1610.09769*, 2016.

[103] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[104] Prachi Singh, Amrit Kaul, and Sriram Ganapathy. Supervised hierarchical clustering using graph neural networks for speaker diarization. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[105] Sharon D Solomon, Kristina Lindsley, Satyanarayana S Vedula, Magdalena G Krzystolik, and Barbara S Hawkins. Anti-vascular endothelial growth factor for

neovascular age-related macular degeneration. *Cochrane Database of Systematic Reviews*, (8), 2014.

[106] Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. Knowledge association with hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2010.02162*, 2020.

[107] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 2015.

[108] Teruhiko Terasawa, Tomas Dvorak, Stanley Ip, Gowri Raman, Joseph Lau, and Thomas A Trikalinos. Systematic review: charged-particle radiation therapy for cancer. *Annals of internal medicine*, 151(8):556–565, 2009.

[109] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[110] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pages 382–392. Springer, 2005.

[111] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[112] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[113] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

[114] Lucas Vinh Tran, Yi Tay, Shuai Zhang, Gao Cong, and Xiaoli Li. Hyperml: A boosting metric learning approach in hyperbolic space for recommender systems. In *Proceedings of the 13th international conference on web search and data mining*, pages 609–617, 2020.

[115] Siw Waffenschmidt, Marco Knelangen, Wiebke Sieben, Stefanie Bühn, and Dawid Pieper. Single screening versus conventional double screening for study selection in systematic reviews: a methodological systematic review. *BMC medical research methodology*, 19(1):1–9, 2019.

[116] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Active learning for biomedical citation screening. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 173–182, 2010.

[117] Byron C Wallace, Thomas A Trikalinos, Joseph Lau, Carla Brodley, and Christopher H Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC bioinformatics*, 11(1):55, 2010.

[118] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Class imbalance, redux. In *2011 IEEE 11th international conference on data mining*, pages 754–763. IEEE, 2011.

[119] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 819–824. ACM, 2012.

[120] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.

[121] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proc. of AAAI*, 2017.

[122] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019.

[123] Tingyi Wanyan, Chenwei Zhang, Ariful Azad, Xiaomin Liang, Daifeng Li, and Ying Ding. Attribute2vec: Deep network embedding through multi-filtering gcn. *arXiv preprint arXiv:2004.01375*, 2020.

[124] Chih-Hsuan Wei, Bethany R. Harris, Donghui Li, Tanya Z. Berardini, Eva Huala, Hung-Yu Kao, and Zhiyong Lu. Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts. *ArXiv preprint*, 20d.

[125] Meng Xiao, Ziyue Qiao, Yanjie Fu, Hao Dong, Yi Du, Pengyang Wang, Hui Xiong, and Yuanchun Zhou. Hierarchical interdisciplinary topic detection model for research proposal classification. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[126] Qianqian Xie, Edward J Schenck, He S Yang, Yong Chen, Yifan Peng, and Fei Wang. Faithful ai in medicine: A systematic review with large language models and beyond. *medRxiv*, 2023.

[127] Yiqing Xie, Zhen Wang, Carl Yang, Yaliang Li, Bolin Ding, Hongbo Deng, and Jiawei Han. Komen: Domain knowledge guided interaction recommendation for

emerging scenarios. In *Proceedings of the ACM Web Conference 2022*, pages 1301–1310, 2022.

[128] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. *arXiv preprint arXiv:1904.07785*, 2019.

[129] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *TKDE*, 2020.

[130] Carl Yang, Jieyu Zhang, and Jiawei Han. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 721–730. IEEE, 2020.

[131] Menglin Yang, Min Zhou, Rex Ying, Yankai Chen, and Irwin King. Hyperbolic representation learning: Revisiting and advancing. *arXiv preprint arXiv:2306.09118*, 2023.

[132] Yaming Yang, Ziyu Guan, Jianxin Li, Jianbin Huang, and Wei Zhao. Interpretable and efficient heterogeneous graph convolutional network. *ArXiv preprint*, 2020.

[133] Ke Yu, Shyam Visweswaran, and Kayhan Batmanghelich. Semi-supervised hierarchical drug embedding in hyperbolic space. *Journal of chemical information and modeling*, 60(12):5647–5657, 2020.

[134] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019.

[135] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 1249–1261, 2021.

[136] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3065–3072, 2020.

[137] Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1106–1117, 2020.