MEDTO: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks

Junheng Hao[1*], Chuan Lei[2], Vasilis Efthymiou[3*], Abdul Quamar[2],
Fatma Özcan[4*], Yizhou Sun[1], Wei Wang[1]

jhao|yzsun|weiwang@cs.ucla.edu,chuan.lei@ibm.com,vefthym@ics.forth.gr,ahquamar@us.ibm.com,fozcan@google.com

[1]University of California, Los Angeles, CA 90095
[2]IBM Research - Almaden, 650 Harry Road, San Jose, CA 95120
[3]FORTH - Institute of Computer Science, Heraklion, Crete, Greece
[4]Google, 1600 Amphitheatre Parkway, Mountain View, CA, 94043

ABSTRACT

Medical ontologies are widely used to describe and organize medical terminologies and to support many critical applications on healthcare databases. These ontologies are often manually curated (e.g., UMLS, SNOMED CT, and MeSH) by medical experts. Medical databases, on the other hand, are often created by database administrators, using different terminology and structures. The discrepancies between medical ontologies and databases compromise interoperability between them. Data to ontology matching is the process of finding semantic correspondences between tables in databases to standard ontologies. Existing solutions such as ontology matching have mostly focused on engineering features from terminological, structural, and semantic model information extracted from the ontologies. However, this is often labor-intensive and the accuracy varies greatly across different ontologies. Worse yet, the ontology capturing a medical database is often not given in practice. In this paper, we propose MEDTO, a novel end-to-end framework that consists of three innovative techniques: (1) a lightweight yet effective method that bootstrap a semantically rich ontology from a given medical database, (2) a hyperbolic graph convolution layer that encodes hierarchical concepts in the hyperbolic space, and (3) a heterogeneous graph layer that encodes both local and global context information of a concept. Experiments on two real-world medical datasets matching against SNOMED CT show significant improvements compared to the state-of-the-art methods. MEDTO also consistently achieves competitive results on a benchmark from the Ontology Alignment Evaluation Initiative.

CCS CONCEPTS

• **Computing methodologies** → *Knowledge representation and reasoning*; **Ontology engineering**; • **Applied computing** → **Health care information systems**.

KEYWORDS

Ontology matching; graph neural network; medical data

1 INTRODUCTION

In recent years, many medical ontologies have been created from various healthcare resources, semi-automatically or by human experts. Medical ontologies define, standardize and organize concepts in the medical domain, which provide valuable knowledge to support many healthcare applications, such as medical content browsing, clinical documentation, and evidence-based healthcare. Examples of medical ontologies include International Classification of Diseases (ICD), Unified Medical Language System (UMLS), and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT). Ontologies are widely used in data integration [8] and query federation to provide standard semantics across multiple systems. They are also used to enhance answers for medical databases using techniques known as Ontology-Based Data Access (OBDA) [41].

Much of the literature in OBDA is devoted to the study of query answering under the assumptions that the ontology is available and the mappings between the database and the standard ontology have already been provided. However, we observe that these assumptions do not necessarily hold in real-life medical applications. A conversational system [29] that we built for the medical database of IBM Micromedex®, used by medical experts (e.g., doctors, nurses, pharmacists), revealed that the database was not designed with a target ontology and there was no mapping between the tables of the database and any known medical ontologies like UMLS or SNOMED CT. We also observe a similar issue on public medical datasets such as MIMIC-III [17], in which some tables and the associated columns are named with abbreviations or colloquial terms. These two use cases show the need for an end-to-end system that maps the medical database to standard medical ontologies, such that the downstream applications (e.g., OBDA) can benefit from the standard terminologies and vocabularies, discover additional relationships, and answer semantically rich queries.

There have been many efforts devoted to ontology matching [11, 16, 20], to find a mapping between two given ontologies. Ontology matching is only part of the problem we are trying to address in this paper, namely *d*ata to ontology matching. When there is no ontology associated with a given database, we need to create an ontology describing the data as a first step, before we can apply ontology matching.

Most of the ontology matching work, such as LogMap [16] and AML [11], rely on logical reasoning and rule-based methods to extract various sophisticated features from the ontologies. These terminological and structural features are then used to compute ontological concept similarities that drive the ontology matching. However, these features in one ontology often do not transfer in others. Consequently, the accuracy and robustness of ontology matching based on different features vary greatly with different medical ontologies to be matched [20]. Worse yet, these solutions assume that the given ontologies are carefully crafted, which often fall short of the requirements for data to ontology matching.

Recently, graph representation learning [14, 19] has emerged as an effective approach to learn vector representations for graph-structured data. The representation of a node is learned by recursively aggregating the representations of its neighboring nodes. Several studies [34, 39, 40] have exploited graph neural networks (GNNs) for embedding-based entity alignment as similar entities usually have similar neighborhoods in knowledge graphs (KGs). Although existing GNN-based methods have achieved promising results on entity alignment in KGs, they are still facing three critical challenges when applied to data to ontology matching.
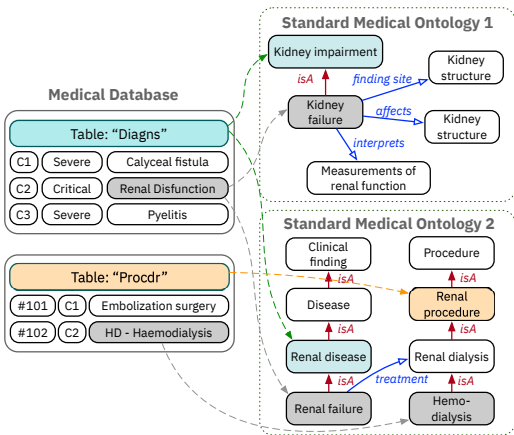

Figure 1: Example of data to ontology matching.

First, data to ontology matching often suffers from a cold-start problem, where a semantically rich ontology capturing a given medical database does not exist. One can generate an ontology from the relational database [24] only using its metadata. However, we argue that if we enrich the ontology by using instance-level information from the database, and incorporate a richer set of semantic relationships, the derived ontology can be matched to the standard ontology with higher precision. To overcome this cold-start problem, a bootstrapping process is necessary.

Second, one distinct characteristic of medical ontologies, compared to the open-domain knowledge graphs like DBpedia [23] and

YAGO [32], is their deep domain specialization. These ontologies often have rich hierarchical top-down structures, which systematically organize medical concepts into categories and subcategories of different levels from general to specific. Figure 1 shows two snippets of medical ontologies. The hierarchical (through "isA" relations) neighborhood of *"kidney failure"* is very different from other types of relations. Capturing such hierarchical structures separately would help identify matching concepts and improve the accuracy of ontology matching.

Third, standard medical ontologies often are non-isomorphic in the local neighborhood structures of a concept from the one of a derived ontology. The rich and complex vocabularies, abundant sources of domain knowledge, and different modeling views all contribute to such non-isomorphism greatly. Fortunately, many medical ontologies have top-level concepts provided by domain experts, and such concepts provide a global context for matching concepts. In Figure 1, the concept *"clinical finding"* is the top-level category of *"renal failure"*. This helps us differentiate *"renal failure"* from other concepts such as *"renal dialysis"*, which belongs to the *"procedure"* category. Motivated by the fact that the semantically related latent information can appear in these top-level concepts, the aggregated neighborhood of a concept should include not only its local neighbors, but also the concepts with its global information.

To cope with these challenges, we propose a <u>me</u>dical <u>d</u>ata <u>t</u>o <u>o</u>ntology matching (MEDTO) framework based on graph representation learning. The underlying idea is to first create and enrich a source ontology from the given medical database, and then embed both enriched and standard medical ontologies into two representations (i.e., hierarchical and non-hierarchical views) that are complementary to each other. Both representations are jointly optimized to improve the ontology matching capabilities. Our contributions are listed as follows:

• We propose an end-to-end framework MEDTO for data to ontology matching. MEDTO first bootstraps an ontology, based on a given medical database, and then learns and unifies hierarchical and non-hierarchical representations of two ontologies for matching.

• We design a lightweight yet effective method to create and enrich an ontology from the metadata of a medical database with rich semantic information from its instance data.

• We employ hyperbolic graph convolution layers to encode the parent and child concepts of each concept in the hyperbolic space, capturing the hierarchical characteristics in an ontology.

• To enrich the features of each concept, we introduce heterogeneous graph layers to incorporate both the local structure and the global context into concept embeddings.

• Our experiments on matching two real-world medical datasets to SNOMED CT show that MEDTO significantly outperforms the state-of-the-art methods. We also evaluate MEDTO on a benchmark from the Ontology Alignment Evaluation Initiative (OAEI), showing that MEDTO consistently achieves state-of-the-art results.

2 PRELIMINARIES AND SYSTEM OVERVIEW

2.1 Graph Neural Networks

Graph neural networks (GNNs) are deep learning based methods that operate on graph-structured data. It has been shown that GNNs are effective for various applications, such as node classification,

link prediction and community detection. A generalized framework [1] of GNNs consists of a graph encoder and a graph decoder, taking as input an adjacency matrix A, as well as optional node and edge features $X = \{X_N, X_E\}$. A typical graph encoder parameterized by Θ_{enc} combines the graph structure with node and edge features to produce node embedding matrix as:

$$Z = \text{ENC}\,(A, X, \Theta_{enc})\,. \tag{1}$$

The graph encoder uses the graph structure to propagate and aggregate information across nodes and learn embeddings that encode local structural information. A graph decoder is often used to compute similarity scores for all node pairs for downstream tasks on node, edge, or graph level.

Depending on the graph properties, a wide variety of GNNs have been developed. Representative examples include a message-passing neural network R-GCN [31] and a metapath-based neural network HAN [38] for heterogeneous graphs, non-Euclidean hyperbolic GCN [2] for hierarchical graphs, and EvolveGCN [27] for dynamic graphs. More details can be found in Section 6.

2.2 Problem Formulation

Definition 2.1. A medical database $\mathcal{D}$ is represented by a relational schema $\mathcal{S}$ and its instance $\mathcal{I}$. A schema is a finite collection of relation symbols. Each relation symbol has a specified arity, which intuitively corresponds to column names. An instance $\mathcal{I}$ over $\mathcal{S}$ is a collection of relations whose arities match those of the relation symbols in S.

Definition 2.2. A medical ontology is represented as $O = (C, \mathcal{R}, \mathcal{T})$, where C is the set of concepts, $\mathcal{R}$ is the set of relations, and $\mathcal{T} = C \times \mathcal{R} \times C$ is the set of triplets.

Problem definition. Given a medical database $\mathcal{D}$ and a standard medical ontology O, the **data to ontology matching problem** is to find matches $\mathcal{M}$ that map the schema $\mathcal{S}$ of $\mathcal{D}$ to O, such that $\{(i, j) \in \mathcal{S} \times O \mid i \equiv j\}$.

Note that a single standard medical ontology may only partially match with a medical database. In this case, multiple medical ontologies can be used to match against the given database in sequence. In essence, the challenges of matching data to ontology remain due to the semantically poor schema of the medical database and the complex structure of the medical ontology. Hence, we need to design an end-to-end system addressing these challenges.

2.3 System Overview

As depicted in Figure 2, we propose a framework, MEDTO, which consists of two phases: **data to ontology bootstrapping** and **ontology to ontology matching**. Given a medical database, the data to ontology bootstrapping phase first derives an ontology from its schema and data instances. It also bootstraps seed matches between the derived and standard ontologies by labeling highly confident matches and adding them into training data. The ontology to ontology matching phase takes as input the derived ontology, the standard ontology, as well as the seed matches (either provided or bootstrapped). Structures of both ontologies are captured via

graph neural networks (GNNs) for structural representation learning. Moreover, the lexical semantics of the concepts in both ontologies is employed, providing complementary signals for ontology matching.
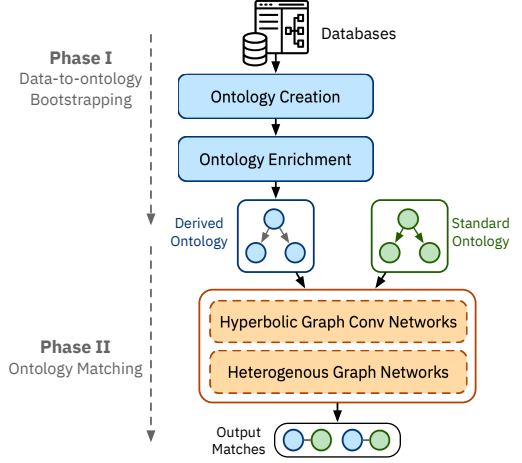

Figure 2: MEDTO system architecture.

3 ONTOLOGY BOOTSTRAPPING FROM MEDICAL DATABASE

In this section, we examine the ontology bootstrapping problem. Specifically, we first address the "cold-start problem", i.e., the task of creating an ontology from a medical database. Then, we describe our novel concept augmentation and neighborhood augmentation strategies to enrich the derived ontology.

3.1 Ontology Creation

To infer an ontology (i.e., concepts and their relationships relevant to the domain) from a relational database, we leverage a variety of information from both database schema and data instances.

Concepts and properties. We map each table in a medical database to a concept and represent columns in each table as data properties of that concept. Note that not all columns are selected, as they may not be semantically meaningful. Specifically, primary and foreign keys are not included, because they are designed for uniquely identifying each row in the table. Moreover, columns of non-string types (e.g., numeric, date, etc.) are not chosen either, since most standard medical ontologies only contain concepts expressed in strings.

Relation inference. Relation inference is non-trivial as it depends on the primary key and foreign key interactions, and quite often these keys are not specified in the databases, especially when the database is created from raw medical literature. Therefore, we follow the approach suggested in [24], which enables the inference of functional relations as well as concept hierarchies (i.e., isA relation).

In brief, we first identify primary and foreign keys by leveraging data statistics, such as distinct values. If the number of distinct values of the column and the total row count in the table are identical, we assert a primary key constraint. Similarly, for foreign keys,

we check if the rows in the join of the two tables based on the selected columns are equal to the total rows of the referring table. Furthermore, we consider tables with exactly two columns, both acting as foreign keys to different tables in the schema, as intermediate tables. For every non-intermediate table R, we generate a functional relation that connects the concept C generated from R to another concept C' generated from the table R', if one of R's column is a foreign key referring to R'. If there is a table R_1 with a single column, which is a foreign key referring to a table R_2, we consider the concept C_1 generated from R_1 as subsumed by the concept C_2 generated from R_2. In this case, we assert an isA relation between two concepts corresponding to these two tables. Finally, the resulting ontology O_1 is stored in OWL2 format.

3.2 Ontology Enrichment

Although the created ontologies capture schema-level details of the underlying data, they are far less semantically rich than the standard ontologies created by experts. To alleviate this issue, we introduce two effective augmentation heuristics to enrich the derived ontology O_1 from the medical database.
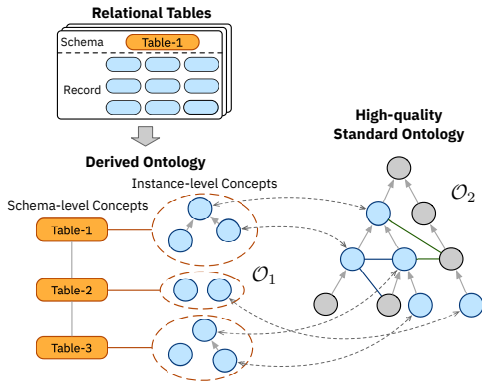

Figure 3: MEDTO ontology enrichment.

Concept augmentation. For each distinct value[1] in relational tables, we add an instance-level concept in O_1, and connect these new concepts to the existing schema-level ones via a new relationship "*instance of*". The advantages of concept augmentation are twofold. First, it greatly enriches O_1 with the available information from the relational database. Second, it enables us to bootstrap the seed concept matching between two ontologies using exact string matching algorithms. Other approximate string matching algorithms (e.g., edit-distance based or embedding based) or ML-based methods [33] can be plugged in as well, depending on the accuracy requirement.
Neighborhood augmentation. We also add edges among the pre-aligned seed concepts in O_1. Specifically, if two concepts i and j of O_2 have an edge, while their counterparts i' and j' in O_1 do not, we add an edge between i' and j'. The goal is to fill the semantic gap between O_1 and O_2 by adding the missing structural information.

With the augmented ontology, MEDTO can effectively learn the ontology representation and align it with O_2. To match a schema-level concept in O_1 with the ones in O_2, we employ graph pooling

---

[1]If the number of distinct values is greater than a threshold, we use sampling to avoid exploding the ontology. We omit the details due to space constraints.

to aggregate the embeddings of instance-level concepts that belong to the schema-level concept. Different graph pooling methods [14, 42] have been investigated for different scenarios. We find that the element-wise *mean*-pooling is sufficient to capture different information across the neighborhood set.

Finally, we feed both the enriched O_1 and a standard ontology O_2 into our novel graph neural network MEDTO to find the matches between them (Figure 3).

4 ONTOLOGY MATCHING

4.1 Input Embeddings of Medical Concepts

The concept names in a medical ontology consist of sequences of words. One can leverage deep learning based embedding methods such as BERT [6] or ELMo [28] to produce d_{in}-dimensional word embeddings for each concept. In this work, as the starting point, we choose BioBERT [22], a high-quality medical language model pre-trained on PubMed abstracts and clinical notes (MIMIC-III). The resulting input embedding is used as the initial state ($h^{0,E}$) of each concept, where E indicates that the embeddings are in a Euclidean space.

4.2 Hyperbolic Graph Convolution Layer

Conventional GNNs embed nodes into Euclidean space, which has been shown to incur a large distortion with hierarchical structures [26]. Hence, we use a hyperbolic embedding space, since it is amenable for learning concept hierarchies. Compared to Euclidean spaces, hyperbolic spaces better capture the hierarchical characteristic of ontologies. In this paper, we adopt a specific model, hyperbolic graph convolutional neural network (HGCN) [2], which leverages both the expressiveness of GNNs and hyperbolic geometry to learn node representations for graphs with hierarchical structures.

Hyperbolic graph convolution layer first establishes mapping between tangent (Euclidean) and hyperbolic spaces by exponential and logarithmic maps. We use the exponential map to project the node embeddings from a Euclidean space to a hyperbolic space, and logarithmic map reverses the map back to the Euclidean space. Hence, the initial embedding $h_i^{0,E}$ of node i to $h_i^{0,H}$ is:

$$\mathbf{h}_i^{0,H} = \exp_o^K \left(0, \mathbf{h}_i^{0,E} \right), \tag{2}$$

where K determines the constant negative curvature $-1/K$ ($K > 0$) and o denotes the origin in the hyperbolic space. For hyperbolic feature transformation from one layer to the next layer, we follow the definition below:

$$\mathbf{h}_i^{l,H} = \left(\mathbf{W}^l \otimes^{K_{l-1}} \mathbf{h}_i^{l-1,H} \right) \oplus^{K_{l-1}} \mathbf{b}^l \tag{3}$$

where $\otimes$ and $\oplus$ are hyperboloid matrix multiplication and addition, respectively, as defined in [2].

Similar to GCN, our hyperbolic graph convolution layer aggregates features from a node's local neighborhood. Since there is no notion of vector space structure in a hyperbolic space, we have to map embeddings to the tangent space, perform the aggregation in the tangent space, and then map the aggregated embeddings back to the hyperbolic space. Furthermore, we utilize an attention mechanism to learn the importance of each neighboring node and aggregate neighbors' embeddings according to their importance. Given hyperbolic embeddings (h_i^H, h_j^H), the attention weight w_{ij}
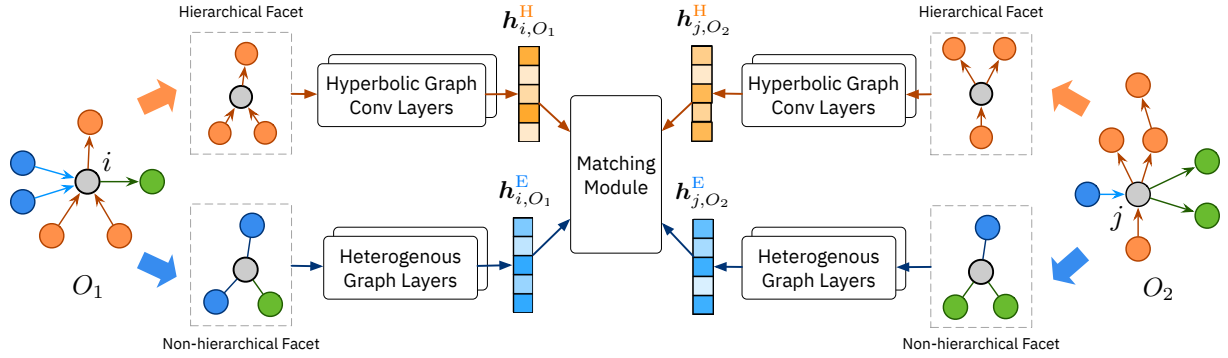
Figure 4: Details of MEDTO matching module. Both O_1 and O_2 are split into the hierarchical and non-hierarchical facets, which are fed into a hyperbolic graph layer and a heterogeneous graph layer respectively. The matching module minimizes the contrastive matching loss to let the representations of matching concepts have a very small distance while those of unmatched concepts have a large distance.

is:

$$w_{ij} = \text{SOFTMAX}\left(\text{MLP}\left(\log_o^K\left(\mathbf{h}_i^H\right) \,||\, \log_o^K\left(\mathbf{h}_j^H\right)\right)\right), \quad (4)$$

and the hyperbolic attention-based aggregation is:

$$\text{AGG}^K(\mathbf{h}^H)_i = \exp_{\mathbf{h}_i^H}^K\left(\sum_{j\in\mathcal{N}(i)} w_{ij}\log_{\mathbf{h}_i^H}^K\left(\mathbf{h}_j^H\right)\right), \quad (5)$$

where $||$ is a concatenation operation, and $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{R}_{\text{isA}}\}$ denotes a set of parents of concepts $i \in C$. Finally, we use a non-linear activation function to learn non-linear transformations by first applying the Euclidean non-linear activation in the tangent space and then mapping back to the hyperbolic space:

$$\sigma^{\oplus^{K_{l-1},K_l}}\left(\mathbf{h}^H\right) = \exp_o^{K_l}\left(\sigma\left(\log_o^{K_{l-1}}\left(\mathbf{h}^H\right)\right)\right). \quad (6)$$

The l-th layer of a hyperbolic graph convolution layer is:

$$\mathbf{h}_i^{l,H} = \sigma^{\oplus^{K_{l-1},K_l}}\left(\text{AGG}^{K_{l-1}}\left(\mathbf{h}^{l,H}\right)_i\right), \quad (7)$$

where $-1/K_{l-1}$ and $-1/K_l$ are the hyperbolic curvatures at the $(l-1)$-th and l-th layer, respectively. The hyperbolic embeddings at the last layer can be used to predict the concept similarity. We use the following sigmoid function [2] to compute probability scores for edges:

$$\mathcal{L}^{\mathcal{H}} = p((c_i, c_j) \in C) = \left\{\exp\left[\frac{1}{t}\left(d^K\left(\mathbf{h}_i^H, \mathbf{h}_j^H\right)^2 - r\right)\right] + 1\right\}^{-1}, \quad (8)$$

where $d^K(\cdot, \cdot)$ is the hyperbolic distance and r and t are hyper-parameters.

4.3 Heterogeneous Graph Module

To capture the non-hierarchical structure in an ontology, conventional GNNs such as R-GCN [31] can be applied, as it models multi-relational graphs. Specifically, R-GCN distinguishes different neighbors with relation-specific weight matrices. In the l-th convolutional layer, each representation vector is updated by accumulating the vectors of neighboring nodes through a normalized sum. Formally,

the l-th layer of R-GCN is:

$$\mathbf{h}_i^{l,E} = \sigma\left(\mathbf{W}_0^l\mathbf{h}_i^{l-1,E} + \sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{W}_r^l\mathbf{h}_j^{l-1,E}\right), \quad (9)$$

where W_0^l is the weight matrix for the node itself and W_r^l is used specifically for the neighbors having relation r, i.e., $\mathcal{N}_i^r$, $\mathcal{R}$ is the relation set and $c_{i,r}$ is for normalization.

One limitation of this approach is that it focuses only on the local context of a concept and ignores the position of the concept within the broader context of the entire ontology. As described in Section 1, the top-level concepts in an ontology often provide additional semantic information which can influence how the final embeddings are aggregated. In Figure 5, the local context of *"renal failure"* includes two concepts, *"measurements of renal function"* and *"kidney structure"*, connecting to *"renal failure"*. In addition to the local context, we also incorporate the "global" context described by the top-level concepts, such as *"clinical finding"*, *"body structure"*, and *"procedure"*.
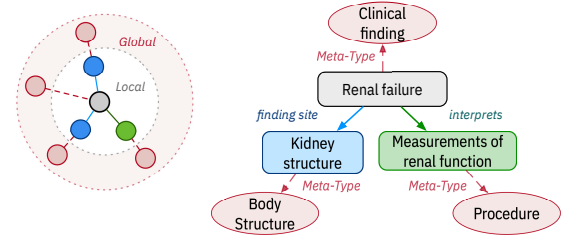
Figure 5: Local and global contexts of "renal failure".

The key idea is to incorporate a set of "global" contexts and enrich each node's feature with its corresponding global embeddings. We denote a node i's global embedding at l-th layer as $\mathbf{g}_i^{l,E}$. We replace the node feature $\mathbf{h}_i^{l-1,E}$ with its enriched version $\mathbf{h}_i^{l-1,E}||\mathbf{g}_i^{l-1,E}$ and similarly replace each node feature of its neighbors $\mathbf{h}_j^{l-1,E}$ with concatenated $\mathbf{h}_j^{l-1,E}||\mathbf{g}_j^{l-1,E}$ in Eq. 9. Note that in a medical ontology, a concept may belong to multiple top-level concepts. In this case,

we take an element wise mean of all global embeddings to fully capture the global context. Such combined embeddings help us to learn better representations from more neighborhood information.

Following the convention, we optimize for cross-entropy loss to push the model to score observable edges higher than the negative ones:

$$\mathcal{L}^{\mathcal{E}} = \sum_{r \in \mathcal{R}} \sum_{i,j \in C} w_{ij}^r \log \frac{\exp\left((\mathbf{h}_i^E)^{\mathsf{T}} A_r \mathbf{h}_j^E\right)}{\sum_{i' \in C} \exp\left((\mathbf{h}_{i'}^E)^{\mathsf{T}} A_r \mathbf{h}_j^E\right)}, \quad (10)$$

where $w_{ij}^r = 1(i, j \in \mathcal{R}_r)$ and negative samples are generated by replacing i with a random node i'.

4.4 Matching Module

Based on the learned concept representations $\mathbf{h}^H$ and $\mathbf{h}^E$ from the hyperbolic graph convolution and the heterogeneous graph layers, we merge the two through concatenation to unify the representation of a concept $\mathbf{h}$. Then, the matching module $M(\cdot)$ takes pairs of concept embeddings from O_1 and O_2 and outputs the prediction score. We use the straightforward multi-layer perceptron (MLP) with one hidden layer, defined as follows:

$$M\left(\mathbf{h}_i^U, \mathbf{h}_j^U\right) = \sigma\left(\mathbf{W}_2 \cdot \gamma\left(\mathbf{W}_1\left(\mathbf{h}_i^U || \mathbf{h}_j^U\right) + \mathbf{b}_1\right) + \mathbf{b}_2\right), \quad (11)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are parameters, σ is the sigmoid function, and γ is the LeakyReLU activation function. One could also use a multi-head attention-based transformer encoder [36] module to replace the MLP.

We minimize the contrastive matching loss to let the embeddings of known matched concepts (positive) have a small distance while the unmatched (negative) pairs have a relatively large distance:

$$\mathcal{L}^M = \sum_{(i,j) \in \mathcal{M}^+} M(\mathbf{h}_i, \mathbf{h}_j) + \sum_{(i',j') \in \mathcal{M}^-} \omega\left[\lambda - M\left(\mathbf{h}_{i'}, \mathbf{h}_{j'}\right)\right]_+, \quad (12)$$

where $\mathcal{M}^+$ denotes the seed matches between O_1 and O_2, $\mathcal{M}^-$ denotes a set of negative samples, λ is the margin value, ω is a balance hyper-parameter, and $[\cdot]_+ = \max(0, \cdot)$.

4.5 Training

Combining the hyperbolic graph convolution and heterogeneous graph models together with the matching module, MEDTO minimizes the final joint loss function:

$$\mathcal{L} = \mathcal{L}^M + \alpha_1 \cdot \left(\mathcal{L}_{O_1}^{\mathcal{H}} + \mathcal{L}_{O_2}^{\mathcal{H}}\right) + \alpha_2 \cdot \left(\mathcal{L}_{O_1}^{\mathcal{E}} + \mathcal{L}_{O_2}^{\mathcal{E}}\right), \quad (13)$$

where $\mathcal{L}^M$ is the matching loss, $\mathcal{L}_{O_1}^H$ ($\mathcal{L}_{O_2}^H$) and $\mathcal{L}_{O_1}^E$ ($\mathcal{L}_{O_2}^E$) represent the losses of the hyperbolic graph convolution and heterogeneous graph models, respectively, and both α_1 and α_2 are positive hyper-parameters to control the trade-off among three loss components. We optimize all models with Adam [18] optimizer.

5 EXPERIMENTS

5.1 Datasets

We use the following datasets from the medical domain to evaluate the performance of our MEDTO framework.

MIMIC-III is a large database consisting of anonymized health-related data of over forty thousand patients who stayed in critical care units [17]. It contains 21 tables in 3 aspects including patient tracking, ICU data, and hospital data.

MDX is a medical database of IBM Micromedex®[2] that contains information in 59 tables about drugs, adverse effects, indications, findings, etc. It is manually curated from medical literature by editorial staff.

For standard medical ontologies, we choose the ones provided in the large BioMed track of OAEI[3]. This track consists of finding alignments between three ontologies: the Foundational Model of Anatomy Ontology (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI).

FMA is an ontology for biomedical informatics that represents a coherent body of explicit declarative knowledge about human anatomy [30]. It consists of 78,984 concepts and 78,985 isA relations.

NCI provides reference terminologies for clinical care, translational and basis research, and public information and administrative activities [5], which consists of 56,907 concepts and 85,332 relations of 80 different types. 59,794 of them are isA relations.

SNOMED CT is a systematically organized collection of medical terms providing codes, terms, synonyms and definitions used in clinical reporting [10]. It contains 76,730 concepts and 109,896 relations, of which 105,563 are isA.

Seed matches are provided by OAEI and we split them into train, validation and test set as the positive samples. The negative samples are uniformly sampled by modifying one of the concepts in the positive sample pairs.

5.2 Compared Methods

To evaluate both phases of MEDTO, we compare our approach against a variety of methods in different categories. For MEDTO ontology bootstrapping phase, we choose the method introduced in ATHENA [15, 24] as the baseline, which only utilizes the schema information from a given database. For MEDTO ontology matching phase, the baselines range from rule-based methods to recent embedding-based entity alignment models. Specifically, LogMap uses logic-based reasoning over the extracted features and casts the ontology matching as a satisfiability problem. AML performs ontology matching based on heuristic methods that rely on aggregation functions. We select MTransE [4], GCN-Align [39], and RDGCN [40][4] from recent embedding-based entity alignment methods. For ablation study, we develop three variants of MEDTO, i.e., MEDTO (w/o HYP) that does not capture the hierarchical information in the hyperbolic space, MEDTO (w/o HET) that does not pay attention to both local and global position information, and the full model MEDTO.

5.3 Implementation Details

The following hyper-parameters are used in the experiments. Each training took 1000 epochs with a learning rate of 0.01. The embedding dimension d is set to 128 for all the comparative methods (if applicable). The dimension of input embeddings is $d_{\text{in}} = 768$. By default, we stack 2 hyperbolic graph convolution and 2 heterogeneous graph layers in MEDTO. For the hyperbolic graph convolution

---

[2]https://www.ibm.com/products/micromedex-with-watson
[3]http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2020/
[4]OpenEA library: https://github.com/nju-websoft/OpenEA

decoder, we set $r = 2.0$, $t = 1.0$ (Eq. 8) and apply trainable curvature. In the matching module, we set $\lambda = 1.0$ and $\omega = 0.1$ (Eq. 12). We set both balance hyper-parameters α_1 and α_2 to 1.0 in Eq. 13. We sample 10 negative samples for each pre-aligned concept pair. All the learnable parameters are initialized by the Xavier initialization [13]. Following the convention of OAEI and entity alignment, we report the precision, recall and F1 score to assess ontology matching performance. In addition, we also report MRR (mean reciprocal rank), higher scores indicating better performance.

5.4 Experimental Results

Main results. We evaluate MEDTO on both MIMIC-III and MDX. For MIMIC-III, our domain experts identified 15 matching concepts in SNOMED, among 21 tables. For MDX, 19 out of 59 tables have their matches identified in SNOMED as well. Hence, we use these identified matches as our ground truth. Following convention, we report Hits@10 and Hits@30 results to assess ontology matching performance.

Table 1: Matching MIMIC-III and MDX to SNOMED CT.

Dataset	MIMIC-III ⇔ SNOMED		MDX ⇔ SNOMED	
Metric	Hits@10	Hits@30	Hits@10	Hits@30
AML	0.06 (1/15)	0.13 (2/15)	0.16 (3/19)	0.26 (5/19)
LogMap	0.20 (3/15)	0.20 (3/15)	0.21 (4/19)	0.37 (7/19)
MTransE	0.00 (0/15)	0.00 (0/15)	0.05 (1/19)	0.05 (1/19)
GCN-Align	0.20 (3/15)	0.33 (5/15)	0.32 (6/19)	0.42 (1/19)
RDGCN	0.27 (4/15)	0.40 (6/15)	0.32 (6/19)	0.58 (11/19)
MEDTO	**0.47 (7/15)**	**0.60 (9/15)**	**0.42 (8/19)**	**0.79 (15/19)**

As shown in Table 1, MEDTO substantially outperforms all baseline methods. For MIMIC-III, the best performing baseline, RDGCN, can only find 4 matches when Hits@10, whereas MEDTO finds 7 out 15 matches. The primary reason is the concept and neighborhood augmentation we used to enhance the initially derived MIMIC-III ontology. With instance-level concepts and hierarchical relationships among them, MEDTO can leverage semantic information to learn much better representations of the MIMIC-III ontology, resulting in the performance gain. We observe similar results on the MDX dataset; our MEDTO finds 8 out of 19 matches compared to 3-6 matches found by other baselines, achieving superior performance. Tables 2 and 3 show a subset of successful and failed cases from both datasets.

Table 2: MIMIC-III-to-SNOMED result analysis (Hits@30).

MIMIC-III Tables	AML	LogMap	RDGCN	MEDTO
patient	✓	✓	✓	✓
prescriptions	✓	✗	✓	✓
caregivers	✗	✓	✗	✓
services	✗	✓	✓	✓
outputevents	✗	✗	✓	✓
icustays	✗	✗	✗	✓
chartevents	✗	✗	✗	✗
labevents	✗	✗	✗	✗

We observe two mistake patterns from MEDTO. The first type of mistakes is caused by ambiguous semantic information. For example, most instance-level concepts of *"chartevent"* are described by

different timestamps, which do not contribute to the bootstrapping of seed matches between MIMIC-III and SNOMED at all. In fact, MEDTO solely relies on the input embedding of *"chartevent"*, which is not sufficient to locate the correct match in SNOMED.

Table 3: MDX-to-SNOMED result analysis (Hits@30).

MDX Tables	AML	LogMap	RDGCN	MEDTO
AdverseEffect	✓	✓	✓	✓
Dosage	✓	✓	✓	✓
DrugFoodInteraction	✗	✓	✓	✓
ContraIndication	✗	✗	✓	✓
DoseAdjustment	✗	✗	✗	✓
DrugRoute	✗	✗	✗	✗

The second type of mistakes still results from instance-level concepts augmented in MIMIC-III. Even though MEDTO is able to leverage these concepts to bootstrap the seed matches, these matches do not locate around the provided ground truth matches in SNOMED. For example, most instance-level concepts of *"labevents"* find their matches (e.g., *"hemoglobin"* and *"cholesterol"*) under *"substance"* concept in SNOMED. Consequently, MEDTO learns an incorrect representation of *"labevents"* and mistakenly matches it to concepts similar to *"substance"* rather than *"laboratory test"*. We observe similar trends from MDX case as well.

Ontology bootstrapping results. As mentioned earlier, we evaluate the effectiveness of MEDTO ontology bootstrapping methods against a baseline method introduced in ATHENA [15, 24], which only utilizes the schema information of a database to create an ontology. MEDTO matches 7 out 15, and 8 out of 19 over MIMIC-III and MDX, respectively, when Hits@10, while ATHENA is only able to match 3 out of 15, and 4 out of 19. Even when Hits@30, ATHENA (4 out of 15 on MIMIC-III, and 5 out of 19 on MDX) is still beaten by MEDTO substantially. The results clearly show that MEDTO ontology bootstrapping method can produce a semantically richer ontology compared to the one generated by ATHENA. Having the enriched ontology, the ontology matching phase can subsequently identify more matching concepts from the standard ontology.

Ontology matching results. Table 4 summarizes the results of ontology matching on three pairs of ontologies from OAEI datasets. We observe that MEDTO outperforms the three representative baselines from entity alignment, with an average improvement of 4.7% on F1 score and 2.5% on MRR. This indicates that entity alignment methods, designated for general-purpose knowledge bases (e.g., Wikidata and DBpedia), are insufficient for matching domain-specific medical ontologies with hierarchical structures. MEDTO explicitly distinguishes and models the hierarchical information, from other local and global structural features, leading to better results on medical ontology matching.

Compared to the extensively developed rule-based approaches (AML/LogMap), MEDTO achieves competitive results across all three datasets. In particular, MEDTO outperforms both AML and LogMap on NCI-SNOMED matching. It is the most challenging one among the three matching tasks, since both NCI and SNOMED are more complex than FMA. We also find that AML and LogMap heavily rely on lexical features from a suite of sophisticated matchers. Deriving such features for a given ontology can be time-consuming. However, these features in one ontology often do not transfer in others. As

Table 4: Results of ontology matching on OAEI datasets.

Datasets	FMA-NCI				FMA-SNOMED				NCI-SNOMED			
Metrics	P	R	F1	MRR	P	R	F1	MRR	P	R	F1	MRR
AML	0.942	**0.899**	**0.920**	–	**0.902**	0.729	0.806	–	0.890	0.744	0.810	–
LogMap	0.916	0.895	0.905	–	0.791	**0.850**	**0.819**	–	0.897	0.732	0.805	–
MTransE	0.627	0.640	0.633	0.416	0.505	0.475	0.490	0.372	0.254	0.378	0.304	0.349
GCN-Align	0.813	0.783	0.798	0.561	0.763	0.729	0.746	0.526	0.745	0.775	0.760	0.467
RDGCN	0.855	0.843	0.849	0.761	0.824	0.752	0.786	0.683	0.852	0.782	0.816	0.679
MEDTO	**0.944**	0.874	0.908	**0.783**	0.871	0.762	0.813	**0.690**	**0.901**	0.802	**0.849**	**0.704**
MEDTO (w/o HYP)	0.867	0.775	0.818	0.724	0.787	0.653	0.714	0.540	0.835	0.759	0.795	0.595
MEDTO (w/o HET)	0.927	0.851	0.887	0.763	0.863	0.747	0.801	0.676	0.881	**0.807**	0.842	0.688

shown in Tables 2 and 3, the accuracy of such approaches varies dramatically depending on the quality of the given ontologies.

Effectiveness of MEDTO heterogeneous graph layer and hyperbolic graph convolution layer. We compare the performance between the proposed MEDTO and its two variations, named MEDTO (w/o HYP) and MEDTO (w/o HET), which only use the heterogeneous graph layers and hyperbolic graph convolution layers, respectively. Results are also shown in Table 2. We observe that full model MEDTO consistently performs the best across three datasets, with an average increase of 2.3% in F-1 score. This is attributed to MEDTO's unified representation, capturing the critical semantic and structural features from multiple facets. It is also interesting to see that MEDTO (w/o HET) outperforms MEDTO (w/o HYP), which indicates that hierarchical information in medical ontologies contains more representative and critical features of ontology matching. Our hyperbolic graph convolution module effectively encodes such information for the matching module.
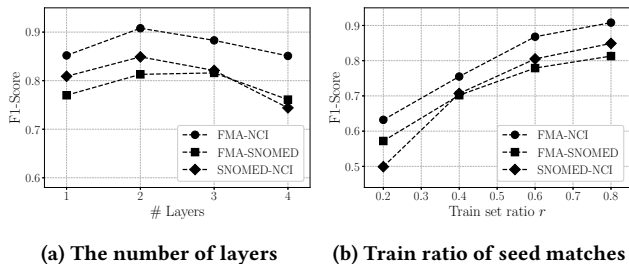

(a) The number of layers   (b) Train ratio of seed matches

Figure 6: Sensitivity analysis.

Hyper-parameter sensitivity analysis. We first analyze the results of MEDTO with 1 to 4 hyperbolic graph convolution and heterogeneous graph layers on OAEI datasets. In Figure 6a, we observe the optimal number of layers is 2 (for FMA-NCI and SNOMED-NCI) or 3 (for FMA-SNOMED). When MEDTO uses more layers, its performance declines. The reason is that MEDTO indirectly captures more global contextual (i.e., top-level concepts) information by message propagation, and such global information would lead to more non-isomorphic neighborhoods.

Furthermore, we also aim to match ontologies with different numbers of seed matches. We use different proportions r of seed matches in OAEI datasets. As shown in Figure 6b, the MEDTO performs substantially better when r increases from 0.2 to 0.4, but

the performance gain slows down as r increases from 0.6 to 0.8. This shows that MEDTO does not heavily rely on a large number of high-quality seed matches and provides decent matching results when the seed matches are limited.

6 RELATED WORK

Graph representation learning. Recently graph representation learning [14, 31] has been intensively studied and shown effective for various tasks including node classification, link prediction and graph matching. Recently, graph attention networks [12, 37, 38] have been introduced and allow each node to attend over its various neighbors and uses attention to assign different weights to different nodes in a neighborhood. We refer interested readers to [9] for more details. Recently, there has been research in extending GNNs to learn non-Euclidean embeddings and thus benefit from both the expressiveness of GNNs and hyperbolic geometry. Poincaré embeddings [26] learn embeddings of hierarchical graphs such as lexical databases (e.g., WordNet) in the Poincare space. HGCN [2] and HGNN [25] apply graph convolutions in hyperbolic space by leveraging the Euclidean tangent space, which provides a first-order approximation of the hyperbolic manifold at a point. These methods lead to improvements on graphs with hierarchical structures.

Ontology matching. Traditional feature-based approaches have been investigated for ontology matching, including terminological-based features, structural-based features and employing external semantic thesauruses for discovering semantically similar entities. LogMap [16] relies on lexical and structural indexes to enhance its scalability. AML [11] also employs various sophisticated features and domain-specific thesauri to perform ontology matching. Feature-based methods mainly employ crafting features to achieve specific tasks. Unfortunately, these hand-crafted features will be limited for a given task and face the bottleneck of improvement.

Representation learning has a recent impact on ontology matching. DeepAlignment [20] is an unsupervised ontology matching system, which refines pre-trained word embeddings with the descriptions of entities, including synonyms and antonyms extracted from general lexical resources and information captured implicitly in ontologies. Similar to DeepAlignment, a framework is introduced for medical ontology alignment [21], based on terminological embeddings. The retrofitted word vectors are learned from the domain knowledge encoded in ontologies and semantic lexicons.

Entity alignment. Similar to ontology matching, entity alignment seeks to find entities in different knowledge graphs (KGs) that

refer to the same real-world object. With the recent success of graph representation learning, embedding-based entity alignment has emerged and attracted massive attention recently [35]. GC-NAlign [39] leverages GCNs for cross-lingual KG alignment. Entity alignments are discovered based on the distances between entities in the embedding space. RDGCN [40] introduces dual relation graphs to capture complex relation information via attentive interaction between KGs. AliNet [34] employs an attention mechanism to key distant neighbors to expand the overlap between entities neighborhood structures. It then controls the aggregation of the direct and distant neighborhood using a gating mechanism. We refer interested readers to the recent survey [35] for more details on embedding-based entity alignment.

Data integration. Much effort has been made to towards data integration [7, 8], including schema alignment and data fusion. Schema mapping methods [3] create a mediated (global) schema and identify the mappings between the mediated (global) schema and the local schemas of multiple databases to determine which attributes contain the same information. Hence, the main goal of data integration is to create the global schema so that multiple databases can be integrated and queried together. In data to ontology, on the other hand, there is only a single database, and the relations in this database are mapped to concepts in a standard ontology to utilize standard vocabularies and enable semantically rich queries.

7 CONCLUSION

In this paper, we propose an end-to-end framework MEDTO for medical data to ontology matching. MEDTO creates a semantically rich ontology from a given medical database and learns multiple facets of concepts in both enriched and standard ontologies. MEDTO encodes the hierarchical information of concepts in the hyperbolic space through hyperbolic graph convolution layers. We further capture both local and global structural information of concepts using heterogeneous graph layers. MEDTO incorporates the information from these layers and learns better concept representations for ontology matching. Our experiments on a variety of real-world medical databases and ontologies demonstrate the effectiveness of MEDTO. As future work, we plan to support different types of matching relations between two concepts (e.g., $\subseteq$, $\supseteq$, and disjoint) and to extend MEDTO to match data to multiple ontologies in a holistic manner.

REFERENCES

[1] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *CoRR*, abs/2005.03675, 2020.

[2] I. Chami, Z. Ying, C. Ré, and J. Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, pages 4869–4880, 2019.

[3] C. Chen, B. Golshan, A. Y. Halevy, W. Tan, and A. Doan. Biggorilla: An open-source ecosystem for data preparation and integration. *IEEE Data Eng. Bull.*, 41(2):10–22, 2018.

[4] M. Chen, Y. Tian, M. Yang, and C. Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517, 2017.

[5] S. de Coronado, M. W. Haber, N. Sioutos, M. S. Tuttle, and L. W. Wright. NCI thesaurus: Using science-based terminology to integrate cancer research results. In *MEDINFO*, volume 107, pages 33–37, 2004.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] A. Doan, A. Y. Halevy, and Z. G. Ives. *Principles of Data Integration*. Morgan Kaufmann, 2012.

[8] X. L. Dong and D. Srivastava. Big data integration. *Proc. VLDB Endow.*, 6(11):1188–1189, 2013.

[9] Y. Dong, Z. Hu, K. Wang, Y. Sun, and J. Tang. Heterogeneous network representation learning. In *IJCAI*, pages 4861–4867, 2020.

[10] K. Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. In *Stud Health Technol Inform*, volume 121, pages 279–290, 2006.

[11] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The agreementmakerlight ontology matching system. In *OTM*, pages 527–541, 2013.

[12] X. Fu, J. Zhang, Z. Meng, and I. King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*, page 2331–2341, 2020.

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[14] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.

[15] M. Jammi, J. Sen, A. R. Mittal, et al. Tooling framework for instantiating natural language querying system. *PVLDB*, 11(12):2014–2017, 2018.

[16] E. Jiménez-Ruiz and B. C. Grau. Logmap: Logic-based and scalable ontology matching. In *ISWC*, pages 273–288, 2011.

[17] A. E. Johnson, T. J. Pollard, L. Shen, et al. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[20] P. Kolyvakis, A. Kalousis, and D. Kiritsis. DeepAlignment: Unsupervised ontology matching with refined word vectors. In *NAACL*, pages 787–798, 2018.

[21] P. Kolyvakis, A. Kalousis, B. Smith, and D. Kiritsis. Biomedical ontology alignment: an approach based on representation learning. *J. Biomed. Semant.*, 9(1):21:1–21:20, 2018.

[22] J. Lee, W. Yoon, S. Kim, et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240, 2020.

[23] J. Lehmann, R. Isele, M. Jakob, et al. Dbpedia-A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 2015.

[24] C. Lei, F. Özcan, A. Quamar, et al. Ontology-based natural language query interfaces for data exploration. *IEEE Data Eng. Bull.*, 41(3):52–63, 2018.

[25] Q. Liu, M. Nickel, and D. Kiela. Hyperbolic graph neural networks. In *NIPS*, pages 8228–8239, 2019.

[26] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, page 6341–6350, 2017.

[27] A. Pareja, G. Domeniconi, J. Chen, et al. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*, pages 5363–5370, 2020.

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[29] A. Quamar, C. Lei, D. Miller, F. Özcan, J. Kreulen, R. J. Moore1, and V. Efthymiou. An ontology-based conversation system for knowledge bases. In *SIGMOD*, pages 361–376, 2020.

[30] C. Rosse and J. L. V. M. Jr. A reference ontology for biomedical informatics: the foundational model of anatomy. *J. Biomed. Informatics*, 36(6):478–500, 2003.

[31] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, et al. Modeling relational data with graph convolutional networks. In *ESWC*, pages 593–607, 2018.

[32] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW*, pages 697–706, 2007.

[33] Z. Sun, W. Hu, Q. Zhang, and Y. Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, page 4396–4402, 2018.

[34] Z. Sun, C. Wang, W. Hu, et al. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*, pages 222–229, 2020.

[35] Z. Sun, Q. Zhang, W. Hu, C. Wang, M. Chen, F. Akrami, and C. Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *PVLDB*, 13(11):2326–2340, 2020.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR*, 2018.

[38] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. Heterogeneous graph attention network. In *WWW*, page 2022–2032, 2019.

[39] Z. Wang, Q. Lv, X. Lan, and Y. Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357, 2018.

[40] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*, pages 5278–5284, 2019.

[41] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev. Ontology-based data access: A survey. In *IJCAI*, pages 5511–5519, 2018.

[42] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. *CoRR*, abs/1806.08804, 2018.