Mathematical Biosciences and Engineering

MBE, 19(8): 8479–8504.
DOI: 10.3934/mbe.2022394
Received: 12 November 2021
Revised: 10 January 2022
Accepted: 01 June 2022
Published: 09 June 2022

http://www.aimspress.com/journal/MBE

Research article

Matching biomedical ontologies with GCN-based feature propagation

Peng Wang[1,2,3,*], Shiyi Zou[2], Jiajun Liu[1] and Wenjun Ke[4]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 210018, China
[2] Monash University Joint Graduate School, Southeast University, Suzhou 215123, China
[3] School of Cyber Science and Engineering, Southeast University, Nanjing 210018, China
[4] Beijing Institute of Computer Technology and Application, Beijing 100854, China

* **Correspondence:** Email: pwang@seu.edu.cn; Tel: +862552090977.

Abstract: With an increasing number of biomedical ontologies being evolved independently, matching these ontologies to solve the interoperability problem has become a critical issue in biomedical applications. Traditional biomedical ontology matching methods are mostly based on rules or similarities for concepts and properties. These approaches require manually designed rules that not only fail to address the heterogeneity of domain ontology terminology and the ambiguity of multiple meanings of words, but also make it difficult to capture structural information in ontologies that contain a large amount of semantics during matching. Recently, various knowledge graph (KG) embedding techniques utilizing deep learning methods to deal with the heterogeneity in knowledge graphs (KGs), have quickly gained massive attention. However, KG embedding focuses mainly on entity alignment (EA). EA tasks and ontology matching (OM) tasks differ dramatically in terms of matching elements, semantic information and application scenarios, etc., hence these methods cannot be applied directly to biomedical ontologies that contain abstract concepts but almost no entities. To tackle these issues, this paper proposes a novel approach called BioOntGCN that directly learns embeddings of ontology-pairs for biomedical ontology matching. Specifically, we first generate a pair-wise connectivity graph (PCG) of two ontologies, whose nodes are concept-pairs and edges correspond to property-pairs. Subsequently, we learn node embeddings of the PCG to predicate the matching results through following phases: 1) A convolutional neural network (CNN) to extract the similarity feature vectors of nodes; 2) A graph convolutional network (GCN) to propagate the similarity features and obtain the final embeddings of concept-pairs. Consequently, the biomedical ontology matching problem is transformed into a binary classification problem. We conduct systematic experiments on real-world biomedical ontologies in Ontology Alignment Evaluation Initiative (OAEI), and the results show that our approach significantly outperforms other entity alignment methods and achieves state-

of-the-art performance. This indicates that BioOntGCN is more applicable to ontology matching than the EA method. At the same time, BioOntGCN substantially achieves superior performance compared with previous ontology matching (OM) systems, which suggests that BioOntGCN based on the representation learning is more effective than the traditional approaches.

Keywords: ontology matching; biomedical ontology; convolutional neural network; graph convolutional network

1. Introduction

In recent years, many biomedical ontologies such as Gene Ontology, Open Biomedical Ontologies (OBO), Unified Medical Language System (UMLS) [1] and Medical Subject Headings (MeSH) have been constructed from various health resources by human experts or semi-automatically. Biomedical ontologies are widely used to describe and organize medical terminologies and to support many critical applications in various fields, such as medical data formats standardization [2], medical or clinical knowledge representation and integration [3], and medical prediction [4]. With the continuous evolution of biomedical data, biomedical terminology is characterized by complexity and ambiguity, which further complicates intelligent biomedical applications. Furthermore, emerging biomedical ontologies are built independently using different terminology and structures, resulting in heterogeneous problems. To implement interoperability across biomedical ontologies, the establishment of meaningful connections between heterogeneous biomedical concepts is critically important [5]. Ontology matching [6,7] is a promising solution to such semantic heterogeneity problems by determining the correspondence between concepts and properties in different biomedical ontologies.

Many ontology matching methods have been proposed [8–10]. Traditional methods can be broadly divided into three categories: 1) terminology-based; 2) structure-based; 3) external knowledge-based. Terminology-based methods are designed to match names or name descriptions of ontology elements. Structure-based methods leverage various types of ontology information, such as names, comments, and structural hierarchies, to compensate for the morphological differences between identical elements [6,11,12]. External knowledge-based methods obtain semantic mappings between syntactically dissimilar ontologies using auxiliary sources, such as taxonomies, dictionaries, and thesauri [9,13,14]. Although these methods work well for ontologies in general domains, they are still confronted with several challenges for matching biomedical ontologies. First, due to the complexity of biomedical terminology, it is difficult for terminology-based methods to distinguish between terms which are textually similar but have completely different meanings. Second, structure-based methods can hardly capture the semantic information in biomedical ontologies that usually contain a large number of complex concepts. Finally, for the external knowledge-based methods, it is difficult to effectively use the knowledge base in the biomedical field.

Recently, some efforts based on deep learning have been devoted to effectively capture ontology features for discovering alignments between biomedical ontologies. Some biomedical ontology matching methods based on deep learning have demonstrated the potential to facilitate the interoperability between ontologies such as DeepAlignment [10], SCBOW+DAE(O) [15]. Moreover, these methods also learn embedding representations of domain knowledge from external resources

such as UMLS to improve the quality of vector representation of concepts or properties. Meanwhile, we note that some knowledge graph embedding methods are proposed to deal with entity alignment (EA), such as MTransE [16], JAPE [17], IPTransE [18], GCN-Align [19], RDGCN [20], and MultiKE [21], etc. Nevertheless, there are several differences between entity alignment and biomedical ontology matching. Firstly, biomedical ontologies usually do not contain entities, and yet there are massive entities when aligning entities in knowledge graphs. This quantitative gap in the number of entities leads to the fact that the EA methods cannot be directly used to match biomedical ontologies. Secondly, the semantic information is rarely utilized in entity alignment methods, while semantic similarity plays an essential role in ontology matching. Thirdly, the unbalanced matching problem is more common in existing ontology matching rather than in entity alignment, which requires to find the matches between an ontology describing a local domain knowledge and another ontology covering the information over multiple domains [22]. Therefore, the EA methods that work well in general-purpose domains are not applicable in the domain of biomedical ontologies. Recently, we observe that several research efforts have drawn attention to this research issue in the biomedical domain. MEDTO [15] uses a hyperbolic graph convolution layer that encodes hierarchical concepts in hyperbolic space and a heterogeneous graph layer that encodes context information of a concept. DAEOM [23] models the matching process by embedding techniques with jointly encoding ontology terminological description and the network structure. Ontoemma [24] develops a neural architecture capable of encoding additional information. However, these works still suffer from the limitation of sparse relational structure and heavy reliance on pre-training and external resources [25].

Recently, graph representation learning has emerged as an effective method for learning vector representations of graph-structured data. High-dimensional graph data are often in irregular forms. They are more difficult to analyze than image/video/audio data defined on regular lattices. Various graph embedding techniques have been developed to convert the raw graph data into a low-dimensional vector representation while preserving the intrinsic graph properties [26]. EPEA [27] uses pairwise connectivity graph for similar feature propagation, which takes full advantage of graph structure and performs well in the EA task. Inspired by this idea, we observe that an ontology can be seen as a graph structure with semantics [28]. Therefore, it is also feasible to apply PCG to ontology matching. This approach is completely different from the traditional methods based on shallow string matching and manual design rules and uses deep learning to learn the ontology graph representation to solve the ontology heterogeneity problem. Specifically, given a source ontology and a target ontology, we first generate a PCG, in which each node is a concept-pair and each edge is a property-pair. In order to learn high quality node embedding, we extract features by a CNN model to obtain short and dense vectors of node representations, and then employ a GCN model to propagate similarity features to obtain the final embedding of nodes.

The major contributions of this paper are as follows:

1) We propose BioOntGCN, a biomedical ontology matching method with GCN-based feature propagation on PCG, whose nodes are concept-pairs and edges are property-pairs. This approach does not require artificially designed rules and can effectively capture the semantic structural information of the ontology through graph neural networks. Furthermore, BioOntGCN converts the ontology matching problem into a binary classification problem by directly learning embeddings of ontology-pairs.

2) We propose a similarity feature extraction method based on convolutional neural network (CNN), which automatically generates feature vectors of concept-pairs or property-pairs to encode

their attribute similarities. This method can automatically obtain useful similarity features of concept-pairs without any human operation.

3) We design a graph convolutional network (GCN) with edge-aware attentions to propagate similarity features in the PCG. Similarity features are propagated among the neighbors of concept-pairs or property-pairs, which incorporate structure similarity into the embeddings of ontology pairs. GCNs learn node embeddings in a graph by recursively aggregating the feature vectors of its neighbors, which are able to combine the node features and structure information in the graph.

4) We conduct the experiments on OAEI datasets. Our approach outperforms the compared approaches and achieves state-of-the-art results.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 formalizes the ontology matching problem. Section 4 describes our proposed approach. Sections 5 and 6 present the evaluation results and Section 7 is the conclusion.

2. Related works

Ontology matching is a rich research field where multiple complementary approaches have been proposed [7,29]. Euzenat and Shvaiko [6] present a comprehensive overview of matching approaches and categorize techniques as terminological, structural, external, and representation learning dimensions. We will focus on discussing related work on ontology matching the biomedical domain.

According to the features used in ontology matching, matching approaches can be classified into four categories: terminology-based approach, structure-based approach, external knowledge-based approach, and representation learning-based approach.

2.1. Terminology-based approach

In the biomedical domain, discovering alignments relying on dictionaries and similarities of terms and labels is a typical ontology matching approach, which is still widely used [6]. Some terminological matchers are exploited as a basic matching method such as ASMOV [14], SAMBO [30], Falcon [19], and AgreementMakerLight [8]. However, the terminology-based approach often provides good precision but a low recall because it is difficult to deal with variations in the form of terms or labels (e.g., equivalence between *hindlimb bone* and *bone of the lower extremity*).

2.2. Structure-based approach

According to the intuition that elements of two distinct ontologies are similar when their adjacent elements are similar, structure-based matchers utilize property attributes and taxonomy hierarchy structure [31]. CroMatcher [32] focuses on the aggregation of distinct matchers in structural level: super-element matcher, sub-element matcher, domain matcher, and range matcher. Similarity flooding [31] presents a structural algorithm based on fixpoint computation and propagation of similarities along with the property relationships between elements that are usable across different scenarios, including biomedical applications. Falcon-AO [33] uses a linguistic matcher combined with a technique that represents the structure of the ontologies to be matched as a bipartite graph. Besides, the similarities between domain elements and between statements in ontologies are computed by recursively propagating similarities in the bipartite graphs. FCA-Map [34] constructs relation-based formal

context to describe the biomedical elements in taxonomic, partonomic, and disjoint relationships with the anchors, and then uses the context to validate the initial lexical mappings. LogMap [9] combines the structural indexation to represent the extended class hierarchy. Contexts for the same anchor are expanded by using the class hierarchies of the input biomedical ontologies to discover new mappings.

2.3. External knowledge-based approach

Matching strategies based on external knowledge provide additional lexical or structural information, allowing for the obtaining of new alignments. Biomedical ontology matching systems explore potential resources or auxiliary knowledge, such as upper-level ontology, WordNet [31], UMLS [1], and BioPortal [35], to find synonyms, spelling variants, and annotations for the concepts to be matched. Systems such as LogMap-Bio [36] and AgreementMakerLight [8] exploit a set of ontologies as background knowledge to generate equivalent mappings. In addition to the anchoring mappings related to the same background ontology, Annane et al. [36] utilize alignments produced by matching intermediate ontology between each other. Faria et al. [37] present a novel approach based on building the specific mapping graph as background knowledge and consider the limitation of the selection and the combination of heterogeneous existing mappings stored in a biomedical repository. It allows getting high-quality alignments between biomedical ontologies without using complex lexical and structural measures.

2.4. Representation learning-based approach

Representation learning is so far rare in ontology matching (OM), particularly in biomedical ontologies. There are a few approaches exploring unsupervised representation learning techniques to capture the interactions among element's descriptions within biomedical ontologies. Zhang et al. [38] investigated the use of representation learning for ontology matching and presented a hybrid method to incorporate word embeddings into the computation of semantic similarities among elements. They were the first that reported that the general-purpose word vectors were not good candidates for the task of ontology matching. Xiang et al. [39,40] proposed an entity representation learning algorithm based on Stacked Auto-Encoders [41,42]. However, training such powerful models with such small training sets is problematic. Wang et al. [24] proposed a neural architecture for biomedical ontology matching called OntoEmma. It encodes a variety of descriptions and derives large amounts of labeled data from biomedical thesaurus for training the model. Considering the problem of distinguishing semantic similarity and descriptive association on rare phrases, Kolyvakis et al. [15] proposed a representation learning method: SCBOW+DAE(O). This approach is a representation framework based on terminological embeddings, in which the refinement of pre-trained word vectors is introduced and learned by the domain knowledge encoded in ontologies and semantic lexicons. However, there still exist the limitations of the sparsity problem of structural relations and heavy dependence on pre-training. MultiOM [43] models the matching process by embedding techniques from multiple views and then optimizes the vector of concepts through a novel proposed negative sampling skill designed for structural relations in biomedical ontology.

3. Problem formulation

In this section, we first formally define the ontology and ontology matching, and then we analyze and compare the differences between ontology matching and entity alignment.

An ontology is composed of triples like $< s, p, o >$ where s, p, and o stand for the subject, predicate, and object, respectively. There are three kinds of ontology resources: uniform resource identifier (URI) resources, literals, and blank nodes. In a triple, the subject can be URIs resources or blank nodes but not literals, and the predicate must be URI resources. Let O be the RDFS (RDF Schema) or OWL (Ontology Web Language) ontology represented by a set of RDF triples T.

An RDF triple *t* ($t \in T$) denotes a statement in the form of $< subject, predicate, object >$. Any node in an RDF triple may be a URI with an optional local name, a literal, or a blank node.

Definition 1. (Ontology) An ontology can be represented as $O = (C, R, I)$, where C, R, and I denote sets of atomic concepts, relations (also named properties), and individuals, respectively. A concept, also known as a class, is a collection of objects with the same properties in a domain. It is defined in RFS and OWL by the predefined properties $rdfs{:}\,Class$ and $owl{:}\,Class$. Properties, also known as relations, are used to express the semantic association between concepts. In both RDFS and OWL, the property p can be denoted as (p_{dom}, p, p_{rng}) or (p_{dom}, p_{rng}), where p_{dom}, p_{rng} is the set of classes, which are called the $Domain$ and $Range$ of p, respectively. In OWL, if the value domain of a property p is a simple data type, which is called a data type property ($owl{:}\,DatatypeProperty$), otherwise it is called an object property ($ObjectProperty$). In RDFS and OWL it is also possible to define parent-child concepts and parent-child properties by means of the properties $rdfs{:}\,subClassOf$ and $rdfs{:}\,subPropertyOf$, thus forming a hierarchy of entities.

For simplicity, the set of concepts and properties is indicated by $\mathcal{E}$.

Definition 2. (Ontology Matching) The matching between two ontologies O and O' is $M = \{m_k | m_k =< e_i, e_j, r, s >\}$, where M is an alignment; m_k denotes a correspondence with a tuple $< e_i, e_j, r, s >$; e_i and e_j represent the expressions which are composed of elements from O and O', respectively; r is the semantic relation between e_i and e_j; r could be equivalence ($=$), generic/specific ($\sqsupseteq$ / $\sqsubseteq$), disjoint ($\perp$), and overlap ($\sqcap$), etc.; and s is the confidence about an alignment and typically in the $[0,1]$ range. Therefore, an alignment M is a set of correspondences m_k.

Figure 1 shows an example of alignments between a mouse anatomy ontology and the NCI Thesaurus. $<$ *hindlimbbone, Boneof LowerExtremity, =, 0.7* $>$ and $<$ *limbbone, Bone f he xtremity, =, 0.8* $>$are equivalent correspondences. In this paper, we only focus on identifying one-to-one equivalence correspondences between two concepts belonging to different ontologies.

As mentioned before, there are noticeable differences between ontology matching and entity alignment, therefore, the methods working well in EA are not directly applicable in the domain of biomedical ontologies. Specifically, we summarize such differences in the Table 1, which are divided into four points, matching elements, semantic relationships, semantic data information and application scenarios. First, in terms of matching elements, the ontology matching usually includes matching property, concept, and instance, while entity alignment tends to consider only entities. In addition, there are a variety of semantic relations in ontologies such as equivalence, disjoint and overlap, etc., whereas entity alignment has only equivalence relation. Meanwhile, ontology matching usually involves more semantics and fewer factual triples, but the entity alignment only contains entity-pairs. Finally, ontology matching and entity alignment also have different application scenarios: the former

provides interoperability between heterogeneous ontologies and the latter finds equivalent entities in KGs and texts.

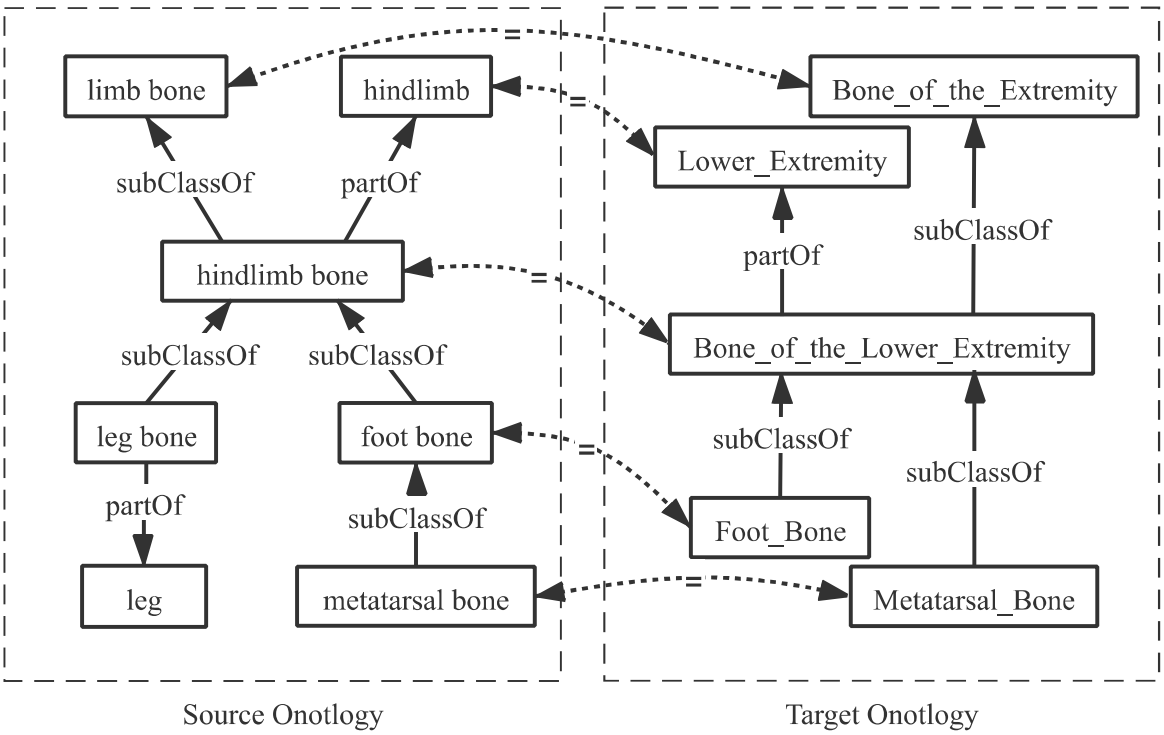

Figure 1. An example of biomedical ontology matching.

Table 1. The differences between ontology matching and entity matching.

	Ontology Matching	Entity Alignment
Matching elements	concept-concept, property-property, instance-instance	entity-entity
Semantic relation	equivalence, generic/specific, disjoint, and overlap, etc.	equivalence
Semantics and data	Rich semantics in ontologies. Few factual triples.	Limited semantics in schema. A mount of factual triples.
Applications	Providing interoperability between heterogeneous ontologies.	Finding same entities in knowledge graphs and texts.

Before introducing the BioOntGCN framework, we give the symbols and fundamental definitions used throughout the paper as Table 2 lists.

Table 2. Symbols and Definition.

Symbol	Definition
s	subject in an ontology.
p	predicate of an ontology.
o	object of an ontology.
O	an ontology.
C	concepts in O.
R	relationships in O.
I	instances or entities in O.
E	the set of entities in O.
M	the alignment of two ontologies.
e	elements from O.
r	the semantic relation.
m_k	the alignment of O and O'.
p_{dom}	the domain of p.
p_{rng}	the range of p.
$\mathcal{R}$	the set of edges in the PCG.
Γ	the set of edge-types in the PCG.
S	similarities of(s_i,s_j), (p_i,p_j) and (o_i,o_j).
A_{data}, A_{obj}	the set of all the data properties and object properties, respectively.
M_{sim}	the similarity matrix of O and O'.
$X^{(l)}$, b_l	the input of l^{th} layer in CNN.
$\tilde{A}$	the adjacency matrix of the undirected graph with added self-connections.
$DNE\ (o, o')$	the normalized edit distance similarity.

4. Methods

In this section, we first introduce the basic framework of our model. Then we will explain the generation of pair-wise connectivity graph. At last, the details of CNN-based Feature Extraction and GCN-based Feature Propagation will be presented.

4.1. Overview of our method

We propose our BioOntGCN framework based on an attention-based feature propagation mechanism. As illustrated in Figure 2, the framework of our approach consists of three components: Pairwise Connectivity Graph, Convolutional Neural Network and Graph Convolutional Network. Our approach first generates a PCG of two ontologies, whose nodes are concept-pairs and edges correspond to property-pairs; it then learns node embeddings of the PCG. To obtain more desirable embeddings, we adopt a Convolutional Neural Network (CNN) to extract similarity features and transform the similarity matrix into a short and dense vector for feature propagation. Further, our approach uses a residual GCN with edge-aware attentions to propagate the property feature, which is built by

modifying the attention mechanism of the GAT model [44]. Finally, a complex ontology matching problem (even an NP-hard problem) is transformed into a binary classification problem.

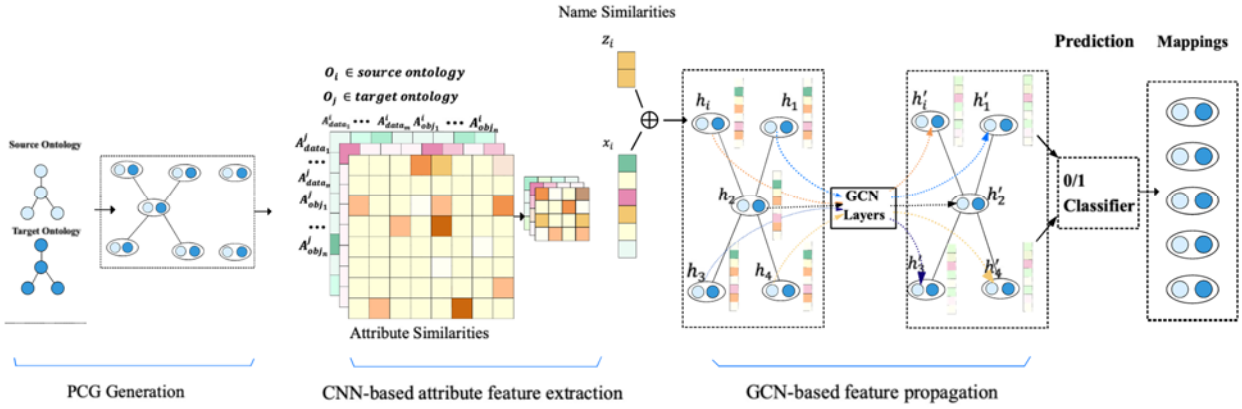

Figure 2. Framework of BioOntGCN.

4.2. Generating the PCG

Pair-wise Connectivity Graph PCG can combine two directed graphs to establish the node-to-node interactions [30,42]. By generating the PCG of two ontologies, the problem of ontology matching is then transformed to node embedding and classification (i.e., equivalent or nonequivalent) in the PCG. In our work, we define the PCG of ontologies. For given two ontologies represented as graph structures, each node in their PCG corresponds to a concept-pair from the two ontologies, and each edge connecting the two nodes reflects the correlation between two concept-pairs. Specifically, for two ontologies O and O', $O = (C, R, I)$ and $O' = (C', R', I')$, the PCG of them is $PCG(O, O')$, which consists of a triple shaped as $< \Omega, \Re, \mathcal{T} >$, where Ω, $\Re$and $\mathcal{T}$ sets of nodes, edges and edge-types. Each element in Ω, corresponds to an ontology-pair between O and O', and each element in $\Re$ corresponds to a relation-pair. Each edge is constructed as follows:

$$<< x, y >, < p, p' >, < x', y' >> \in PCG(O, O') \Leftrightarrow\Rightarrow < x, p, y > \in O, < x', p', y' > \in O' \quad (1)$$

Figure 3 shows an example of PCG of from $O \times O'$. There are two ontologies, each of them has three concepts. The PCG of them contains nine nodes representing all possible concept-pairs of two ontologies; and there are five typed edges in the PCG. PCG can represent the connections of concept-pairs between two ontologies, we use PCG to capture the interaction of possible concept alignments between two ontologies. In our approach, the problem of ontology matching will be solved via node embedding of the PCG. Equivalent relations of concepts are predicted based on the learned embeddings.

To generate the PCG of two ontologies, we can first pair all the concepts and properties from two ontologies as nodes, and then use Eq (1) to generate edges between nodes. In fact, we run into the problem that the PCG generated from large-scale biomedical ontologies contain a large number of useless node pairs. To overcome this issue, we adopt the propagation strength condition (PSC), which can effectively select concept-pairs having high equivalent possibilities as node in the PCG. Before

using PSC, we first use one of the following methods to generate initial representations of concepts, which are used in PSC.

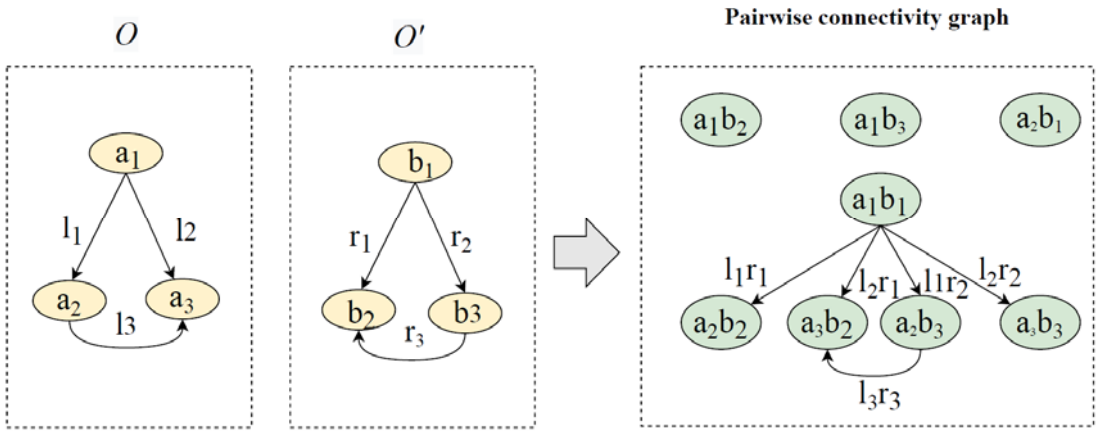

Figure 3. Pair-wise connectivity graph.

• **N-grams of Concepts.** Generating a set of character-level n-grams of concepts as the set-representations.

• **N-grams of Properties.** This method treats property of an ontology as text strings and generates character-level n-grams of all the properties for each ontology. All the n-grams are then merged into a set as the representation of the ontology.

• **Credible Initial Seeds.** To provide better initial similarity seeds for similarity calculation and propagation, these initial seeds can be selected and generated by other matching methods.

Propagation Strength Condition Given two triples $t_i =< s_i, p_i, o_i >$ and $t_j =< s_j, p_j, o_j >$, and let S_s, S_p and S_o denote the corresponding similarities of (s_i, s_j), (p_i, p_j) and (o_i, o_j), respectively. Similarities can be propagated only t_i and t_j satisfy following three conditions:

• In S_s, S_p and S_o, at least two similarities must be larger than threshold θ;

• If t_i includes ontology language primitives, the corresponding positions of t_j must be same primitives;

• t_i or t_j has at most one ontology language primitive.

Condition 1 ensures that the final similarity results are creditable after propagating. The ontology language primitives refer to RDF and OWL vocabularies. Condition 2 ensures that two triples use same ontology language primitive to describe semantics. For example, *<Conference_paper, rdfs:subClassOf, Paper>* and *<Paper, rdfs:subClassOf, Document>* use the RDF primitive *rdfs:subClassOf* as predicate, so the similarities can be propagated between them. Condition 3 ensures that there is no ontology definition and declaration triples during propagating, because such triples may cause incorrect matching results. For example, without condition 3, two triples *<PhDStu, rdf:type, rdfs: Class>* and *<Paper, rdf:type, rdfs:Class>* will cause wrong alignment: *PhDStu = Paper.*

4.3. Ontology feature extraction

Generally, concepts with the same or similar properties or parent-child concepts tend to be equivalent. Therefore, the properties of ontology are considered to be the key terms to discover

ontology alignments. In traditional approaches, properties have to be first matched manually. Especially in some ontology embedding-based methods, properties are utilized to generate property representations, which are integrated with structure embeddings of concepts to get more accurate ontology matching. In our work, we extract similarity features from properties in an automatic way.

CNN-based Feature Extraction We propose a property feature extraction method based on convolutional neural network. This method can automatically obtain useful similarity features of concept-pairs without any human operation. It generates a vector representation of each concept-pair in the PCG, which captures property similarities of two concepts.

Given a concept-pair $< o_i, o_j >$, where $o_i \in O$ and $o_j \in O'$. Let $A^i_{data} = \{A^i_{data_1}, \dots, A^i_{data_m}\}$ and $A^j_{data} = \{A^j_{data_1}, \dots, A^j_{data_m}\}$ be two sets of all the data properties in O and O', respectively. Let $A^i_{obj} = \{A^i_{obj_1}, \dots, A^i_{data_n}\}$ and $A^j_{obj} = \{A^j_{obj_1}, \dots, A^j_{obj_n}\}$ be two sets of all the object properties in O and O', respectively.

Biomedical ontologies rarely contain instances; hence property values are missing. To keep simplicity and effectiveness, our approach treats all the properties as strings. Similarities of properties are computed as N-gram-based Jaccard similarities of strings:

$$Jaccard(o, o') = \frac{|NG(o) \cap NG(o')|}{|NG(o) \cup NG(o')|} \tag{2}$$

where $NG(s)$ and $NG(t)$ are n-grams of strings o and o'.

Usually, one biomedical ontology concept is described by a small number of properties. Therefore, the similarity matrix of a concept-pair is usually a sparse one, with a large proportion of 0s in it. Meanwhile, similarities between some properties may be useless for detecting ontology alignments. The proposed CNN model solves this problem by automatically and efficiently extracting the property features of concept-pairs, encoding the sparse similarity matrix into a dense and relatively short vector.

The input of the CNN is the similarity matrix M_{sim} of two concepts, two convolution layers are used to generate a dense similarity vector from M For the l^{th} convolution layer, its output is computed as follows:

$$X_k^{(l)} = ReLU(W_k^{(l)} \otimes X^{(l-1)} + b_k^{(l)}) \tag{3}$$

where $X^{(l-1)}$ is the input of l^{th} layer; for the first layer, $X^{(0)} = M$; we use multiple filters to extract useful similarity features from the input, $X^{(l)}$ is k thek^{th} filter of l^{th} layer, $b^{(l)}$ is the bias of the k^{th} filter in l^{th} layer; $\otimes$ is the convolution operator. There is a max pooling layer after each convolution layer. The output features of last max pooling layer are the similarity vector of the concept-pair. In addition, we choose the ReLU function as the activation function, which outputs non-linear results, reflecting the effect of hidden layers. Furthermore, it can effectively avoid the problem of gradient disappearance due to deep CNN and save a lot of computation when back-propagating.

Label Similarity Features The label of a concept is generally seen as an important clue to determine whether two ontologies are matched. Therefore, our work treats the label as a special property, computes a label similarity vector for each concept-pair, which will be concatenated with the similarity vector generated by the CNN model. To capture similarity features of concepts' labels from

different aspects, we use multiple string-based similarity metrics, which are widely used in traditional similarity-based alignment approaches. The following similarity measures are used in our approach:

• **Edit Distance.** It evaluates the minimal cost of operations which have to applied to one of the strings to obtain the other string:

$$z_1(o, o') = 1 - \frac{|ops|}{max(len(o), len(o'))} \tag{4}$$

where $|ops|$ denotes the set of operations, $len(\cdot)$ is the string length.

• **Edit Similarity.** It denotes the edit similarity between two strings:

$$z_2(o, o') = \frac{|o|+|o'|-z_1(o,o')}{2} \tag{5}$$

where $|o|$ denotes the string length.

• **Jaccard Similarity.** It computes the Jaccard Similarity of the character-level n-grams of two strings, as defined in Eq (2), we denote this similarity as $z_3(o, o')$.

However, in some ontologies, the local names of elements are represented in the form of ID, such as *NCI_12269* in NCI Thesaurus and *MA_0000216* in MA ontology, which is meaningless. Consequently, we first simply obtain the mapping results through comparing the label sets of the pairs of elements. The normalized edit distance similarity metric is applied to compute linguistic similarities between label sets:

$$SIM_{name} = DNE(o, o') = \frac{z_1(o,o')}{z_1(o,o')+z_2(o,o')} \tag{6}$$

The normalized edit distance similarity is denoted as $DNE(o, o')$. After that, mapping results are generated through a given threshold filtering and similarity ranking. This is an empirical threshold obtained from experimental data on biomedical ontologies. Therefore, it remains consistent across all data sets of our experiments. Specifically, we determined this threshold to be 0.35.

Let SIM_{name} denote label similarities of an ontology-pair, it will be concatenated with the similarity vector x generated by CNN to form the initial feature vector of the ontology-pair. The feature vectors of all the ontology-pairs will be passed to a propagation process, to generate the final embeddings.

4.4. Feature propagation

Neighbors of equivalent ontologies are usually also equivalent or similar. Therefore, structure information in ontologies is crucial for discovering ontology alignments. In our work, edges between nodes in the PCG reflect the neighboring information of concept-pairs. To obtain feature representations of concept-pairs containing their neighbors' information, our approach propagates property features of concept-pairs following these edges. Specifically, our approach uses a graph convolution network (GCN) to propagate the property features of concept-pairs over the PCG. GCNs learn node representations in a graph by recursively aggregating the feature vectors of its neighbors, which are able to combine the node features and structure information in the graph. Several approaches have exploited GNNs for embedding-based KG alignment, which achieved promising results. In

previous approaches, GNNs are used for learning representations of individual entity or concept. While in this work, we design a new GCN model for learning vector representations of concept-pairs.

Graph Convolution Networks GCNs [45,46] are neural networks operating on unlabeled graphs and including features of nodes based on the structures of their neighborhoods. We consider a multi-layer graph convolutional network (GCN) with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{l}W^{l}) \tag{7}$$

Here, $\tilde{A} = A + I_N$is the adjacency matrix of the undirected graph with added self-connections. I_N is the identity matrix, $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W_{(l)}$ is a layer-specific trainable weight matrix. I_N denotes an activation function, such as the $ReLU(\cdot) = max(0,\cdot)$. $H_{(l)} \in \mathbb{R}^{N\times D}$ is the matrix of activation in the l^{th} layer; $H^{(0)} = X$.

Our model is a residual GNN with edge-aware attentions, which is built by modifying the attention mechanism of the GAT model [40]. Our GNN model has two layers, each layer takes a set of node features $H = \{h_1, h_2, \dots, h_N\}$ as inputs, where $h_i \in R^F$ and N is the number of nodes in the PCG, F is the dimension of the input features. Each layer generates a new set of node representations $H' = \{h_1{}', h_2{}', \dots, h_N{}'\}$, $h_i{}' \in R^{F'}$ and it is computed as:

$$h_i' = \sigma\big(\sum_{j\in N_I} \alpha_{ij} W h_j\big) \tag{8}$$

where N_i is the set of neighbouring nodes of the $i-th$ node (ignoring the edge directions in the PCG), W $\in R^{F\times F'}$ is a shared matrix, α_{ij} is a learnable attention indicating the importance of the $j-th$ node to the $i-th$ node.

Edge-aware Attention Mechanism In the GAT model, the attention α_{ij} is computed based on the features of node i and j.

In the task of ontology matching, we consider that the type of edge between two nodes is important and should not be ignored. Therefore, we use an edge-aware attention mechanism to compute the attention α_{ij}.

A shared attention mechanism $R^{F'} \times R^{F'} \times R^{F'} \to R$ is used to computes attention coefficients:

$$e_{ij} = LeakyReLU\big(a^T[Wh_i \parallel Wh_j \parallel t_{(i\to j)}]\big) \tag{9}$$

where $(i \to j)$ denotes the index of edge-type linking the $i-th$ node to the $j-th$ node, $t_{(i\to j)} \in R^{F'}$ is the vector representation of edge-type; $a \in R^{3F'}$ is the vector of an edge of a single-layer feed-forward neural network for computing the attention coefficients; $\parallel$ denotes concatenation of vectors. Here the vector of an edge of an edge-type is computed by it. For an edge-type t_k, let S_k and T_k be the sets of nodes' indices having outgoing edges and coming edges of the type in the PCG respectively, the vector representation of t_k is computed as:

$$t_k = |\frac{1}{|S_k|}\sum_{i\in S_k} Wh_i - \frac{1}{|T_k|}\sum_{j\in T_k} Wh_j| \tag{10}$$

which is the element-wise absolute difference between the mean vectors of source and target nodes connected by t_k.

When the attention coefficients are obtained following Eq (9), normalized attentions are then computed using a softmax function over all the coefficients of its neighbouring nodes:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})} \tag{11}$$

where N_i is the set of neighbouring nodes of the *i-th* node.

4.5. Model training

Model training is divided into two main parts, CNN model training for automatic extraction of property features and GCN model training for feature propagation. These two separate models are trained sequentially, using the same training data i.e., already aligned ontologies. For the CNN model, let x_i be the property feature vector of ontology-pair (e_i, e_j) generated by the model. We use one fully-connected layer to generate a score for each ontology-pair, taking x_i as the input:

$$S_{CNN}(m, n) = \sigma(c^T x_i + \alpha) \tag{12}$$

where $c \in R^d$ and $\alpha \in R$ are parameters, σ is the sigmoid function.

For the GCN model, let h_i be the feature vector of the ontology-pair (e_i, e_j) after the feature propagation with the model.

A similar score function is also defined as:

$$S_{GCN}(m, n) = \sigma(Th_i + \beta) \tag{13}$$

where $g \in R^d$ and $\beta \in R$ are parameters, σ is also the sigmoid function.

Both models are trained by minimizing the following margin-based ranking loss function:

$$\mathcal{L} = \sum_{(m,n) \in A} \sum_{(m',n') \in A'} [\gamma - S(m, n) + S(M', N')]_+ \tag{14}$$

where $[x] = max\{0, x\}$, $\gamma > 0$ is a margin hyper-parameter, $A'_{(m,n)}$ denotes the set of non-aligned ontology-pairs in the PCG containing ontology m or n. The score S is either S_{CNN} or S_{GCN}, depending on which model is trained.

At the same time, we note that full graph training on PCG graphs with node sizes of up to a million would not work well. Consider an L-layer graph neural network with hidden state size H running on an N-node graph. Storing the intermediate hidden states requires $O(LNH)$ memory, consuming a large amount of computing resources. To solve this problem, we design the algorithm of PCGblocking training.

TransformRootless Algorithm We take as input a PCG in the form of a triplet, and first we check whether there is a closed loop in this graph structure, and if there is no closed loop in PCG then it is identified as a tree structure. Next, we check whether PCG has multiple root nodes, i.e., whether it is a rooted tree. For multiple root nodes, the similarity between nodes is calculated using edit distance, and if it exceeds the threshold 0.035, it is fused into one node to realize the transformation of unrooted tree to rooted tree. Algorithm 1 shows the algorithm for transforming PCG graph to rooted tree structure.

Algorithm 1. TransformRootless algorithm

Input: PCG
Output: PCG with rooted structure

```
Function TransformRootless(PCG)
begin
    if not IsClosedLoop(PCG) then
        if IsRootlessTree(PCG) then
            foreach n_i ← topNodesSets do
                foreach n_j ∈ topNodeSet and n_i ≠ n_j do
                    if (Lev(n_i, n_j) > 0.035)
                        merge(n_i, n_j)
                    end
                end
            end
        end
    end
Function IsClosedLoop(PCG)
begin
    foreach n_i ∈ PCG do
        n_i ← labeled
        foreach n_ij ∈ Neigh(n_i) do
            if n_ij is labeled then
                return False
                break
                else
                  n_ij ← labeled
            end
            return True
        end
    end
end
Function IsRootLessTree(PCG)
begin
    topNodeNum ← 0
    foreach n_i ∈ PCG do
        if n_i has no supClass and topNodeNum ⩾ 1 then
            return True
            else
              continue
        end
        return False
    end
end
```

SerializedBlock Algorithm We take the PCG which is transformed into a rooted tree structure as input and sort the nodes in the graph by depth-first search. Each S nodes are divided into chunks for subsequent chunking training, where S is obtained by the following equation:

$$S = \sqrt{N} \tag{15}$$

where N is the number of nodes in PCG. Algorithm 2 presents the process of serialization blocking.

Algorithm 2. SerializedBlocking algorithm

Input: Rooted PCG
Output: Blocked PCG
Function *SerializedBlocking* (PCG)
begin
    DepthFirstSerilization (PCG)
    Blocking (PCG)
end
Function *Blocking* (PCG)
begin
    i ← 0
    blockNum ← $\sqrt{len(PCG)}$
    wihle (i < $len(PCG)$) do
        $merge(n_i, n_{i+1}, \ldots n_{i+\ blockNum})$
        i ← i+ blockNum
    end
end

5. Experiments

In this section, we provide details of the experiments, i.e., the datasets used, baseline models, experimental setup, results and their analysis including ablation study.

5.1. Datasets

Our experiments are conducted on four ontologies that appear in the Ontology Alignment Evaluation Initiative (OAEI). Two of them (the Adult Mouse Anatomy Ontology and the Foundational Model of Anatomy) are pure anatomical ontologies, while the other two (SNOMED-CT and NCI Thesaurus) are broader biomedical ontologies.

Adult Mouse Anatomy is a structured dictionary that provides standardized nomenclature for anatomical terms in the postnatal mouse and organizes anatomical structures for the postnatal mouse spatially and functionally [47].

Foundational Model of Anatomy (FMA) is an evolving computer-based knowledge source for biomedical informatics. The FMA is a domain ontology of the concepts and relationships that pertain to the structural organization of the human body [47].

NCI Thesaurus (NCI) provides reference terminology for many NCIs and other systems. It covers vocabulary for clinical care, translational and basic research, public information, and administrative activities [48].

SNOMED-CT is a systematically organized computer-processable collection of medical terms providing codes, terms, synonyms, and definitions used in clinical documentation and reporting [49].

Table 3. Summary statistics of the biomedical ontology matching tasks.

Task and ontology	Concepts	Labels	Synonyms	Properties	Triples
Anatomy					
MA	2737	3084	344	2	15,958
NCI	3298	9403	5236	1	35,354
FMA-NCI					
FMA	3696	9142	0	24	16,919
NCI	6488	17,109	0	63	64,857
FMA-SNOMED					
FMA	10,157	26,986	0	24	47,730
SNOMED	13,412	13,431	0	18	110,029

We provide some details regarding the respective size of each ontology matching task on Table 3. For Anatomy, there are 2737 concepts in source ontology and 3298 concepts in target ontology, simultaneously including many labels and synonyms but only the $PART\_OF$ property with both ontologies. FMA-NCI task selects a small part of FMA and NCI ontology, with 3696 concepts from FMA and 6488 concepts from NCI, and FMA-SNOMED also selects a fragment of these ontologies with tens of thousands of concepts, 10,157 concepts in the source ontology FMA and 13,412 concepts in the target ontology SNOMED. For FMA-NCI and FMA-SNOMED, there exists no synonym within the ontologies but some properties to define the relations between entities, 24 properties for FMA, 63 properties for NCI, and 18 properties for SNOMED. For each concept, there are almost several aliases (labels) that are important for the alignments of heterogeneous ontologies. The evaluation of tasks is summarized through the Matching Evaluation Toolkit (MELT) framework supported in OAEI. Actually, the alignments of tasks FMA-NCI and FMA-SNOMED are conducted on a small fragment of the aforementioned ontologies.

5.2. Measures

In order to measure the performance of the matching system, we selected precision, recall, and F-measure adapted for ontology matching evaluation.

We compare the mapping Map, which consists of all those correspondences generated by our system, against reference mapping Ref to compute precision $precision$, $recall$, and F1-measure F. The standard measures for evaluating mappings are denoted as follows:

$$precision(Map, Ref) = \frac{|Map \cap Ref|}{Map} \tag{16}$$

$$recall(M, R) = \frac{|Map \cap Ref|}{Ref} \tag{17}$$

$$F - Measure(M, R) = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{18}$$

In addition, we also consider a comparison with some KGE methods and used Hit@k and MRR (Mean Reciprocal Ranking) as the evaluation metrics, which are popular and widely used in other KG alignment work.

- **Hit@k.** It measures the percent-age of correctly alignments ranked in the top k candidates.
- **MRR.** It is the average of the reciprocal ranks of the results.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{19}$$

where Q is the sequence of returned matches, the correct match ranked in $i - th$ position of the sequence.

5.3. Experiments settings

We implement our approach by using Pytorch and conduct the experiments on a computer with an Intel Xeon 4110 CPU and 64-GB memory. The dimensions of similarity features and final embeddings of concept-pairs are set to the same value, which is among $\{30,60,100,120\}$, we consider the learning rate in two models among $\{0.1,0.01,0.002,0.001\}$, the margin γ in loss functions among $\{1,2,4,10\}$. The best experimental parameter settings of our approach will be given according to the MRR.

We compare our approach with recent KG alignment models, which can be divided into two groups. Models in one group is based on structure information in KGs, including MTransE [16], IPTransE [18], and RDGCN [20], etc. Another is based on similarity, including GCN-Align [19], JAPE [17], and MultiKE [21], etc.

6. Results

6.1. Comparisons of KGEs

Table 4 shows the results of some recent approaches based on the structure of information. The parts of the results separated by solid line denote TransE-based methods and GCNs-based methods.

For TransE-based methods, even TransEdge and BootEA, which have excellent performance in entity alignment, do not perform well in biomedical ontology matching tasks. This is because that the way to contextualize and translate them into entity embeddings between entity pairs in terms of specific head-to-tail entity pairs is not applicable to ontology alignment. TransEdge achieved the best performance of all TransE-based approaches, our approach gets improvements of 71.3, 30.3, and 53.4% of Hits@1 on these tasks.

For GCN-based methods, GCN-Align performs worse due to simple utilization relation triples. We note that RDGCN achieved the best performance on three tasks no account of CEA, our approach gets improvements of 39.2, 21.1, and 36.3% of Hits@1 on these tasks. Although GCN is able to capture

more structural characteristics of knowledge graphs, especially when using more GCN layers, it is still not enough for a small number of properties and instances in biomedical ontology. The good performance of BioOntGCN is largely attributed to its capability for learning relation-aware concept-pair embeddings.

Table 4. Results of KGE alignment based on structure information.

Method	Anatomy			FMA-NCI			FMA-SNOMED		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MTransE	0.211	0.407	0.339	0.268	0.346	0.236	0.119	0.328	0.261
JAPE	0.225	0.357	0.403	0.204	0.334	0.375	0.193	0.208	0.234
BootEA	0.337	0.692	0.588	0.502	0.614	0.547	0.333	0.421	0.478
TransEdge	0.256	0.498	0.432	0.311	0.609	0.586	0.253	0.436	0.402
IPTransE	0.251	0.664	0.429	0.223	0.237	0.178	0.154	0.324	0.336
GCN-Align	0.264	0.289	0.305	0.251	0.327	0.334	0.193	0.239	0.301
RDGCN	0.532	0.607	0.447	0.349	0.533	0.296	0.158	0.234	0.320
HGCN	0.403	0.511	0.448	0.427	0.533	0.598	0.337	0.402	0.463
DGMC	0.472	0.677	0.583	0.637	0.544	0.574	0.359	0.536	0.599
CEA	0.656	0.698	0.543	0.554	0.601	0.577	0.438	0.497	0.380
BioOntGCN	0.876	0.953	0.921	0.807	0.856	0.844	0.754	0.833	0.805

6.2. Comparisons of similarity-based systems

Table 5 illustrates the performances of approaches using attributes or name information compared with our model. Unsurprisingly, the performance of our approach outperforms almost the compared approaches. In terms of Anatomy tracks, DOME [50] and XMap [51] achieve the best performance on precision and recall, respectively, while our method improves the recall and precision by 28.7 and 5.2%, respectively, compared to the two. Furthermore, LogMap [9] and AML which utilize external knowledge provide supplementary lexical or structural information, allowing for the obtaining of new alignments to have better results on FMA-NCI and FMA-SNOMED than several other methods. Our model has an improvement of 5.4, 17.6, and 5.0% on precision, recall and F-measure of NCI-SNOMED task, respectively. It is noteworthy that our approach performs 5.3% better than XMap on precision of FMA-NCI.

6.3. Comparisons of ontology matching systems

Table 6 reports the precision, recall and F-measure of several systems involved in the Ontology Alignment Evaluation Initiative (OAEI) in recent years. We can observe that SCBOW performs well ahead of the competition on FMA-NCI and FMA-SNOMED tasks. Compared with it, our approach improves by 5.4% on precision of FMA-NCI task. For NCI-SNOMED, LogMap outperforms other systems, and our approach has an improvement of 5.4, 17.6, and 4.9% on precision, recall and F-measure, respectively.

Table 5. Results of matchers based on similarity calculation.

Method	Anatomy			FMA-NCI			FMA-SNOMED		
	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.
XMap	0.93	0.87	0.90	0.88	0.74	0.80	0.72	0.61	0.66
DOME	0.99	0.62	0.76	0.80	0.67	0.73	0.94	0.20	0.33
AML	0.96	0.93	0.94	0.81	0.88	0.84	0.69	0.71	0.70
LogMap	0.92	0.85	0.88	0.87	0.81	0.84	0.81	0.64	0.72
FCAMapX	0.94	0.80	0.86	0.67	0.84	0.74	0.82	0.76	0.79
BioOntGCN	0.94	0.87	0.95	0.93	0.79	0.85	0.91	0.77	0.83

Table 6. The comparison of BioOntGCN with OAEI top-ranked systems.

Method	Anatomy			FMA-NCI			FMA-SNOMED		
	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.
AML	0.96	0.93	0.94	0.81	0.74	0.84	0.69	0.71	0.70
LogMap	0.92	0.85	0.88	0.87	0.67	0.84	0.81	0.64	0.72
LogMapBio	0.89	0.90	0.89	0.83	0.88	0.83	0.83	0.65	0.73
Wiktionary	0.96	0.75	0.84	0.60	0.81	0.71	0.78	0.22	0.34
ATBox	0.99	0.67	0.80	0.70	0.84	0.69	0.80	0.21	0.33
Lily	0.90	0.90	0.90	0.90		0.82	0.48	0.52	0.50
ALIN	0.90	0.72	0.83	-		-	-	-	-
SBOW	-	-	-	0.90		0.90	0.86	0.86	0.87
BioOntGCN	0.93	0.94	0.94	0.93	0.79	0.85	0.91	0.77	0.83

6.4. Runtime comparison

Table 7 shows the runtimes of the top-ranked systems that have participated in the OAEI recently. The average training time of our model is four hours and thirty-six minutes, and the average prediction time is 3.23 seconds. It shows that our model greatly accelerates the runtime. To Wiktionary and ALIN as examples, the prediction time on Anatomy are 184 and 811 times faster than these two, respectively. Compared with LogMap, ATMatcher and LogMapBio that are respectively the fastest systems on several tracks of OAEI2021, BioOntGCN achieves an average speed increase of 2.59×, 6.33× and 7.5× for Anatomy, FMA-NCI and FMA-SNOMED.

6.5. Ablation study

In this section, an ablation study is carried out to investigate the necessity of each of the described components, as well as their effect on the ontology matching performance on Table 8.

We set up two sets of variations of BioOntGCN by removing or replacing GCN model. The first variation of BioOntGCN is represented as CNN, which only uses the CNN model to predict alignments based on property features. The second variation of BioOntGCN is represented as GCN, which removes the CNN in BioOntGCN. It shows that two sub-models are both effective and important for the promising performance of our approach. First, the CNN model can extract useful similarity features for predicting for predicting alignments, which gets better results than half of the comparison systems, including ATBox, Wiktionary, etc. There are improvements of 8.8, 10.8, and 10.1% of precision, recall and F1-measure on average when only CNN is used. Second, GCN-based feature propagation

improves the results significantly, because information on neighboring nodes and structures is important for matching. When only GCN is used, there are improvements of 6.2, 5.5, and 4.5% of precision, recall and F1-measure on average.

Table 7. Runtime comparison with OAEI systems.

Method	Anatomy Time(s)	FMA-NCI Time(s)	FMA-SNOMED Time(s)
AML	32	44	124
LogMap	7	24	95
ATMatcher	146	19	30
LogMapBio	-	1190	1434
Wiktionary	493	13435	-
ATBox	-	-	-
Lily	430	-	-
ALIN	2190	-	-
SBOW	-	-	-
BioOntGCN	2.7	3	4

Table 8. Feature ablation study of our proposed approach.

Method	Anatomy			FMA-NCI			FMA-SNOMED		
	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.	Prec.	Rec.	F-m.
BioOntGCN (CNN)	0.89	0.72	0.80	0.85	0.72	0.78	0.79	0.75	0.77
BioOntGCN (GCN)	0.91	0.88	0.81	0.87	0.75	0.81	0.82	0.77	0.81
BioOntGCN	0.93	0.94	0.94	0.93	0.79	0.85	0.91	0.81	0.83

7. Discussion

In this section, we discuss our experimental results according to research questions. First, we will analyze the influence of the matchers in a single dimension. Second, we will report on how information imposes an effect on the final performance compared with the distinguishing clues of concepts.

7.1. How is the impact of seed alignment

To investigate how the size of seed alignments (pre-aligned entity pairs for training) affects the results of our approach, we run our approach with a different number of seed alignments. The proportions of seed alignments range from 5% to 30% with step of 5%. Figure 4 shows the F-measure on three datasets Anatomy, FMA-NCI and FMA-SNOMED. It shows that gets nearly optimal Hits on these datasets when using 30% seed alignments, since our approach can fully utilize property and structure information to accurately predict alignments even with small number of seed alignments although the properties in the ontology are not as abundant as those of the entities.

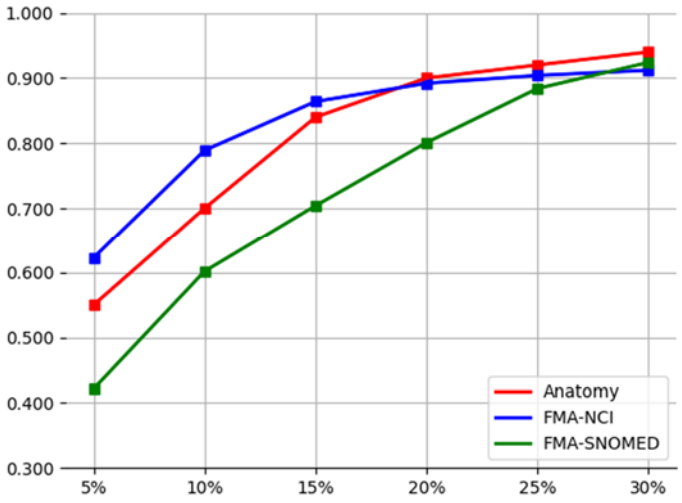

Figure 4. Results of BioOntGCN using different sizes of seed alignments (horizontal coordinates: proportions of pre-aligned concepts used in training data; vertical coordinates: F1-measure).

7.2. How is the performance of properties matching

The performance of alignment systems on property matching lags significantly behind that on class and instance matching. Previously, we conducted an analysis of the performance of string similarity metrics on ontology alignment tasks [52]. One of the findings of that work was that string metrics perform much worse on properties than on classes.

7.3. Why is the bio-Bert-based feature similarity matrix not considered

There are two main reasons for this. On the one hand, the distribution of the bio-Bert word vectors is tapered and uneven in space [53]. High-frequency words are closer to the origin, while low-frequency words are far from the origin, which leads to a huge difference in frequency even if a high-frequency word is semantically equivalent to a low-frequency word, so that the distance of the word vector does not represent the semantic relevance of the words well. On the other hand, the distribution of high-frequency words is compact, and the distribution of low-frequency words is sparse, and the sparse distribution leads to poorly defined semantic intervals and inadequate training of low-frequency word representations.

8. Conclusions

In this paper, we address the problem of biomedical ontology matching from a representation learning perspective, which has been traditionally studied under the setting of feature engineering. We first generate a pair-wise connectivity graph of two ontologies. Then our method learns node embedding of the PCG, which are used to predict matched concepts or properties. CNN is first used

to extract the ontology attribute features, and then we use GCN to propagate the graph structure information from the PCG to obtain embeddings of ontology-pair. Finally, the complex ontology matching problem is transformed into a classification problem. Experimental results demonstrate significant performance gains over the state-of-the-art. Compared with traditional ontology matching methods, our method avoids manual design rules by automatically extracting feature matrices through CNNs. In comparison with current deep learning-based ontology matching methods, the matching effect is improved by making full use of semantic structure information through graph neural networks.

Acknowledgments

The work was supported by the 13th Five-Year All-Army Common Information System Equipment Pre-Research Project (Grant Nos. 31514020501, 31514020503).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. O. Bodenreider, The unified medical language system (UMLS): Integrating biomedical terminology, *Nucleic Acids Res.*, **32** (2004), D267–D270. https://doi.org/10.1093/nar/gkh061
2. J. Cimino, X. Zhu, The practical impact of ontologies on biomedical informatics, *Yearb. Med. Inf.*, **15** (2006), 124–135. https://doi.org/10.1055/s-0038-1638470
3. D. Isern, D. Sánchez, A. Moreno, Ontology-driven execution of clinical guidelines, *Comput. Methods Programs Biomed.*, **107** (2012), 122–139. https://doi.org/10.1016/j.cmpb.2011.06.006
4. P. Potter, H. Cools, K. Depraetere, G. Mels, P. Debevere, J. Roo, et al., Semantic patient information aggregation and medicinal decision support, *Comput. Methods Programs Biomed.*, **108** (2012), 724–735. https://doi.org/10.1016/j.cmpb.2012.04.002
5. Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, Y. Qu, Multi-view knowledge graph embedding for entity alignment, preprint, arXiv:1906.02390.
6. J. Euzenat, P. Shvaiko, *Ontology Matching*, 2nd edition, Springer, 2007. https://doi.org/10.1007/978-3-540-49612-0
7. C. Rosse, J. Mejino Jr, A reference ontology for biomedical informatics: the foundational model of anatomy, *J. Biomed. Inf.*, **36** (2003), 478–500. https://doi.org/10.1016/j.jbi.2003.11.007
8. D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. Cruz, F. Couto, The agreementmakerlight ontology matching system, in *Proceedings of Confederated International Conferences: CoopIS, DOA-Trusted Cloud*, (2013), 527–541. https://doi.org/10.1007/978-3-642-41030-7_38
9. E. Jiménez-Ruiz, B. Grau, Logmap: Logic-based and scalable ontology matching, in *International Semantic Web Conference*, (2011), 273–288. https://doi.org/10.1007/978-3-642-25073-6_18
10. P. Kolyvakis, A. Kalousis, D. Kiritsis, Deepalignment: Unsupervised ontology matching with refined word vectors, **1** (2018), 787–798. https://doi.org/10.18653/v1/n18-1072
11. S. Bergamaschi, S. Castano, M. Vincini, D. Beneventano, Semantic integration of heterogeneous information sources, *Data Knowl. Eng.*, **36** (2001), 215–249. https://doi.org/10.1016/S0169-023X(00)00047-1

12. D. Embley, D. Jackman, L. Xu, Attribute match discovery in information integration: Exploiting multiple facets of metadata, *J. Braz. Comput. Soc.*, **8** (2002), 32–43. https://doi.org/10.1590/S0104-65002002000200004
13. J. Gracia, V. Lopez, M. d'Aquin, M. Sabou, E. Motta, E. Mena, Solving semantic ambiguity to improve semantic web based ontology matching, in *the 2nd International Workshop on Ontology Matching*, (2007), 1–12.
14. Y. Jean-Mary, E. Shironoshita, M. Kabuka, Ontology matching with semantic verification, *J. Web Semant*., **7** (2009), 235–251. https://doi.org/10.1016/j.websem.2009.04.001
15. P. Kolyvakis, A. Kalousis, B. Smith, D. Kiritsis, Biomedical ontology alignment: An approach based on representation learning, *J. Biomed. Semant.*, **9** (2018), 1–20. https://doi.org/10.1186/s13326-018-0187-8
16. M. Chen, Y. Tian, M. Yang, C. Zaniolo, Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, preprint, arXiv:1611.03954.
17. Z. Sun, W. Hu, C. Li, Cross-lingual entity alignment via joint attribute-preserving embedding, in *International Semantic Web Conference*, **10587** (2017), 628–644. https://doi.org/10.1007/978-3-319-68288-4_37
18. Z. Wang, J. Yang, X. Ye, Knowledge graph alignment with entity-pair embedding, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2020), 1672–1680. https://doi.org/10.18653/v1/2020.emnlp-main.130
19. Z. Wang, Q. Lv, X. Lan, Y. Zhang, Cross-lingual knowledge graph alignment via graph convolutional networks, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (2018), 349–357. https://doi.org/10.18653/v1/d18-1032
20. Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, D. Zhao, Relation-aware entity alignment for heterogeneous knowledge graphs, preprint, arXiv:1908.08210.
21. Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, Y. Qu, Multi-view knowledge graph embedding for entity alignment, preprint, arXiv:1906.02390.
22. Q. Zhong, H. Li, J. Li, G. Xie, J. Tang, L. Zhou, et al., A gauss function-based approach for unbalanced ontology matching, in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, (2009), 669–680. https://doi.org/10.1145/1559845.1559915
23. J. Wu, J. Lv, H. Guo, S. Ma, DAEOM: A deep attentional embedding approach for biomedical ontology matching, *Appl. Sci.*, **10** (2020), 7909. https://doi.org/10.3390/app10217909
24. L. Wang, C. Bhagavatula, M. Neumann, K. Lo, C. Wilhelm, W. Ammar, Ontology alignment in the biomedical domain using entity definitions and context, in *Proceedings of the BioNLP 2018 Workshop*, (2018), 47–55. https://doi.org/10.18653/v1/w18-2306
25. P. Wang, Y. Hu, S. Bai, S. Zou, Matching biomedical ontologies: Construction of matching clues and systematic evaluation of different combinations of matchers, *JMIR Med. Inf.*, **9** (2021), e28212. https://doi.org/10.2196/28212
26. F. Chen, Y. C. Wang, B. Wang, C. C. J. Kuo, Graph representation learning: A survey, *APSIPA Trans. Signal Inf. Process.*, **9** (2020), 1–21. https://doi.org/10.1017/ATSIP.2020.13
27. Z. Wang, J. Yang, X. Ye, Knowledge graph alignment with entity-pair embedding, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2020), 1672–1680. https://doi.org/10.18653/v1/2020.emnlp-main.130
28. P. Wang, B. Xu, Matching weak informative ontologies, *Sci. China Inf. Sci.*, **64** (2021), 1–2. https:/ /doi.org/10.1007/s11432-020-3214-2

29. S. Song, X. Zhang, G. Qin, Multi-domain ontology mapping based on semantics, *Clust. Comput.*, **20** (2017), 3379–3391. https://doi.org/10.1007/s10586-017-1087-x
30. P. Lambrix, H. Tan, SAMBO—a system for aligning and merging biomedical ontologies, *J. Web Semant.*, **4** (2006), 196–206. https://doi.org/10.1016/j.websem.2006.05.003
31. G. Miller, WordNet: A lexical database for English, *Commun. ACM*, **38** (1995), 39–41. http://doi.acm.org/10.1145/219717.219748
32. M. Gulić, B. Vrdoljak, M. Banek, Cromatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment, *J. Web Semant.*, **41** (2016), 50–71. https://doi.org/10.1016/j.websem.2016.09.001
33. W. Hu, Y. Qu, Falcon-AO: A practical ontology matching system, *J. Web Semant.*, **6** (2008), 237–239. https://doi.org/10.1016/j.websem.2008.02.006
34. M. Zhao, S. Zhang, W. Li, G. Chen, Matching biomedical ontologies based on formal concept analysis, *J. Biomed. Semant.*, **9** (2018), 1–27. https://doi.org/10.1186/s13326-018-0178-9
35. N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, et al., BioPortal: Ontologies and integrated data resources at the click of a mouse, *Nucleic Acids Res.*, **37** (2009), W170–W173. https://doi.org/10.1093/nar/gkp440
36. E. Jiménez-Ruiz, B. C. Grau, V. Cross, LogMap family participation in the OAEI 2017, in *CEUR Workshop Proceedings*, **2032** (2017), 153–157.
37. D. Faria, C. Pesquita, E. Santos, I. Cruz, F. Couto, Automatic background knowledge selection for matching biomedical ontologies, *PloS One*, **9** (2014), e111226. https://doi.org/10.1371/journal.pone.0111226
38. Y. Zhang, X. Wang, S. Lai, S. He, K. Liu, J. Zhao, et al., Ontology matching with word embeddings, in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, Springer, Cham, (2014), 34–45. https://doi.org/10.1007/978-3-319-12277-9_4
39. M. Sun, H. Zhu, R. Xie, Z. Liu, Iterative entity alignment via joint knowledge embeddings, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, (2017), 4258–4264. https://doi.org/10.24963/ijcai.2017/595
40. X. Xue, A compact firefly algorithm for matching biomedical ontologies, *Knowl. Inf. Syst.*, **62** (2020), 1–17. https://doi.org/10.1007/s10115-020-01443-6
41. Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, 19 (2006), 4–7.
42. A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, *J. Mach. Learn. Res.*, **15** (2011), 215–223.
43. W. Li, X. Duan, M. Wang, X. Zhang, G. Qi, Multi-view embedding for biomedical ontology matching, in *Proceedings of the 14th International Workshop on Ontology Matching collocated with the 18th International Semantic Web Conference*, **2536** (2019), 13–24.
44. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, preprint, arXiv:1710.10903.
45. D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarelli, T. Hirzel, A. Aspuru-Guzik, et al., Convolutional networks on graphs for learning molecular fingerprints, in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, (2015), 2224–2232.

46. S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: Moving beyond fingerprints, *J. Comput. Aided Mol. Des.*, **30** (2016), 595–608. https://doi.org/10.1007/s10822-016-9938-8
47. T. Hayamizu, M. Mangan, J. Corradi, J. Kadin, M. Ringwald, The adult mouse anatomical dictionary: A tool for annotating and integrating data, *Genome Biol.*, **6** (2005), 1–8. https://doi.org/10.1186/gb-2005-6-3-r29
48. J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler, B. Parsia, The national cancer institute's thesaurus and ontology, *J. Web Semant.*, **1** (2003), 75–80. https://doi.org/10.1016/j.websem.2003.07.007
49. S. Schulz, R. Cornet, K. Spackman, Consolidating SNOMED CT's ontological commitment, *Appl. Ontol.*, **6** (2011), 1–11. https://doi.org/10.3233/AO-2011-0084
50. H. Sven, H. Paulheim, DOME results for OAEI 2018, in *Proceedings of the 13th International Workshop on Ontology Matching Collocated with the 17th International Semantic Web Conference*, **2288** (2018), 144–151.
51. X. Xue, J. Zhang, Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm, *Appl. Soft Comput.*, **106** (2021), 107343. https://doi.org/10.1016/j.asoc.2021.107343
52. M. Cheatham, P. Hitzler, String similarity metrics for ontology alignment, in *Proceedings of the Twelfth International Semantic Web Conference*, (2013), 294–309. https://doi.org/10.1007/978-3-642-41338-4_19
53. B. Li, H. Zhou, J. He, M. Wang, Y. Yang, On the sentence embeddings from pre-trained language models, preprint, arXiv:2011.05864.

AIMS Press