Ontology matching using multi-head attention graph isomorphism network

Samira Oulefki[1, 3], Lamia Berkani[2, 3 [0000-0003-3601-2698]], Nassim Boudjenah[3], Imad Eddine Kenai[3] and Aicha Mokhtari[1, 3]

[1] Laboratory for Research in Intelligent Informatics, Mathematics and Applications (RIIMA)
[2] Laboratory for Research in Artificial Intelligence (LRIA)
[3] Dep. of Artificial Intelligence and Data Sciences, Faculty of Informatics, USTHB University, Bab Ezzouar 16111, Algiers, Algeria

{soulefki,lberkani,amokhtari}@usthb.dz

Abstract. Ontology matching is a widely used solution to the semantic heterogeneity problem in data integration or sharing. It consists of establishing mappings between entities that belong to different ontologies. However, as the number of ontologies is increasing for a given domain and the overlap between ontologies grows proportionally, it becomes crucial to develop more reliable and accurate techniques for the automation of this task. While traditional ontology mapping approaches are based on string metrics and structure analysis, some recent methods are using deep neural networks. In this article, we propose a novel approach for ontology matching based on Graph Neural Networks (GNN) as graph representations are helpful for entity and graph comparisons. Our approach is more precisely based on Multi-Head Attention Graph Isomorphism Network (MHAGIN). The results of experiments demonstrate the effectiveness of our approach compared with existing methods.

Keywords: Ontology matching, deep learning, graph neural network, attention mechanism, multi-head attention, graph isomorphism network.

1 Introduction

Over the last decade, ontologies are providing a shared understanding of common domains to meet the need for knowledge sharing between people and/or systems [1]. They provided a mechanism for representing concepts and their relationships within a domain or between different domains. However, given the large number and the variety of ontologies developed (by different developers) for a given domain, how to manage the heterogeneity of ontologies has attracted a considerable attention in recent years. An effective solution to the semantic heterogeneity problem is known as ontology matching [2]. This process refers to the task of establishing correspondences between semantically related entities (i.e. classes / properties) from different ontologies.

Most ontology matching works, such as LogMap [3] and AML [4], relies on logical reasoning and rule-based methods to extract various sophisticated features from ontologies. These terminological and structural features are then used to calculate the similarities between ontological entities that determine ontology mappings. However, features from one ontology are often not transferred to other ontologies. Recently, deep learning (DL) has been widely used in various fields, including ontology matching. Zhang et al. [5] were the first to apply deep learning techniques to ontology matching. The proposed strategy is based on concept similarity. This method based on pre-trained semantic embeddings provided the basic idea for other researches such as OntoEmma [6] and VeeAlign [7]. Some recent approaches are using graph neural networks (GNN) such as BioHAN [8], a GNN-based ontology matching framework with an attention mechanism. However, we noticed a lack of models using graph isomorphism network (GIN) [9], which was proposed as a powerful GNN for graph classification. Accordingly, we propose in this article a novel DL-based ontology matching approach using a multi-head attention graph isomorphism network. Experiments were conducted using two different datasets.

The remainder of this article is organized as follows. Section 2 presents the related work about ontology matching using DL techniques in general and GNN in particular. Section 3 provides the preliminary attached to our context. Section 4 proposes our method for ontology matching. In Section 5 experiments are presented and discussed. Finally, the conclusion highlights the most important results with some perspectives.

2 Related work

The most popular rules-based matching systems are LogMap [3] and AML [4], which have been highly ranked on the Ontology Alignment Evaluation Initiative (OAEI[1]) tracks. By considering two input ontologies, LogMap built their lexical indexing and efficiently calculated initial anchor mappings, i.e. exact mappings. The anchor mappings were then used as a starting point for discovering additional mappings. AML is a scalable ontology matching system, which directly uses an external ontology as a mediator between input ontologies.

Recently, DL has been widely used in various fields, including ontology matching. LogMap or AML output mappings are often used as training data to train supervised methods. Zhang et al. [5] applied DL techniques to ontology matching based on the concept similarity and Natural Language Processing (NLP) techniques that have been integrated to enhance the semantic information of concept embeddings. VeeAlign [7], uses a supervised DL approach to discover alignments. In particular, it uses a two-step model of attention combined with multi-faceted context representation to produce contextualized representations of concepts, which aids matching based on semantic and structural properties of an ontology. Bento et al. [10] used convolutional neural networks to perform string matching between class labels using character embeddings. To further improve the alignment the authors rely on the set of super-classes. He et al. [11] proposed the BERTMap approach that supports both semi-supervised and unsupervised configurations. BERT model is used as it can learn robust contextual embeddings and

[1] http://oaei.ontologymatching.org

requires few resources for fine-tuning. The corpus used is composed of pairs of synonymous labels and pairs of non-synonymous labels. The classifier consists of a linear layer that takes as input the token embeddings produced by BERT's output layer and transforms them into two-dimensional vectors.

Recently, graph representation learning has emerged as an effective method for learning vector representations of graph-structured data. Moreover, as an ontology can be seen as a graph structure with semantics, thus some initiatives are applying GNN for ontology matching. For instance, Wang et al. [12] proposed the BioOntGCN approach that directly learns embeddings of ontology-pairs for biomedical ontology matching through two steps: (1) a convolutional neural network to extract the similarity feature vectors of nodes; and 2) a graph convolutional network to propagate the similarity features and obtain the final embeddings of concept-pairs. Wang and Hu [8] proposed BioHAN through a hybrid graph attention network. First, an ontology-enriching method is proposed to refine and enrich the ontologies through axioms and external resources. Then, a hyperbolic graph attention layers is used to encode hierarchical concepts in a unified hyperbolic space. Finally, the authors aggregated the features of both the direct and distant neighbors with a graph attention network. However, even if these methods were competitive with the state-of-the-art ontology matching methods, we noticed the lack of GIN-based matching models. This motivates us to explore the use of this type of GNN for improving mappings between ontologies.

3 Preliminaries

Graph neural network (GNN). GNN is a type of neural network architecture specifically designed to operate on graph-structured data. It uses the inherent structure of graphs and node features to learn representation vectors of nodes. The process performs through iterative updates where in each iteration, the representation of a node is refined by combining the representations of its neighboring nodes. After k-iterations of this aggregation and updating process, a node's representation encapsulates the structural information present in its k-hop neighborhood [9]. We can formalize the update representation on a node, $\boldsymbol{v}$, in the $\boldsymbol{l}$- th layer as follows:

$$h_v^{(l)} = \mathrm{COMBINE}(AGG^{(l)}\left(\left\{m_u^{(l)}, u \epsilon N(v)\right\}\right), h_v^{(l-1)}) \quad (1)$$

Where $h_v^{(l)}$is the feature vector of the node v in the l- th layer, $COMBINE$ and AGG are the combination and the aggregation functions respectively and $N(v)$ is the set of nodes that are directly attached to node v, i.e., its neighborhood. Formally, $N_v = \{u\colon (u,v)\epsilon\ E$ or $(v,u)\epsilon\ E\}$ and E is the set of edges in the graph.

GNNs have seen several versions and advancements in these recent years. They differ mainly in their implementation of the aggregation and the node representation update functions. In Xu et al [9], the authors analyzed the ability of the variants of GNNs to capture different graph structures. Their results show that the existing GNNs cannot learn to distinguish graph structures. Subsequently, they proposed a new architecture called Graph Isomorphism Networks (GIN) that is as powerful as the Weisfeiler-Lehman graph isomorphism test [9].

Weisfeiler-Lehman test. It is used to determine if two graphs are isomorphic, meaning they have the same structure, but their nodes may have different labels. The Weisfeiler-Lehman (WL) algorithm is an iterative procedure that refines the node labels of a graph by aggregating the neighborhoods of each node and assigning a new label based on the aggregated information. If thealgorithm converges to the same node labeling for both graphs, they are considered as isomorphic. Otherwise, if there is a divergence in the node labeling, the graphs are deemed non-isomorphic [9].

Graph Isomorphism Network (GIN). GIN [9] is designed to be provably the most expressive among the class of GNNs. It is also considered as powerful as the Weisfeiler-Lehman graph isomorphism test. It applies a multi-layer perceptron (MLP) on node features and uses summation as aggregation. The node representation updating formula of GIN is as follow:

$$h_v^{(l)} = \mathrm{MLP}^{(l)}((1 + \epsilon^{(l)}).\, h_v^{(l-1)} + \sum_{u \epsilon N(v)} h_u^{(l-1)}) \tag{2}$$

Where ϵ is a learnable parameter. GIN has traditionally been used for graph classification, but it provably achieves excellent performance for node classification tasks [13].

Ontology as graph. An ontology $\boldsymbol{O}$ can be represented with a directed graph such as: $\boldsymbol{O} = (\boldsymbol{C}, \boldsymbol{E}, \boldsymbol{R})$ where $\boldsymbol{C}$, which is the set of concepts, is represented by the nodes' set of the graph; $\boldsymbol{E}$, the set of relations holding between the concepts, is represented by the edges' set of the graph and $\boldsymbol{R}$ is the set of types of relations.

Ontology Matching. In this era of ever-expanding data and interconnected systems, the need to effectively integrate and harmonize information from multiple ontologies has become necessary. To enable using simultaneously several ontologies, the most common approach consists of creating mappings between their entities. The process allowing to find this mapping is called ontology matching. Formally, we can define ontology matching as a function $\boldsymbol{f}$ taking as input two ontologies and outputs mappings that represent a set of the correspondences holding between their concepts. It is worth noting that in this paper, we consider only the "is-a" relation type between concepts.

4 Our approach

As shown in Fig. 1, our proposed approach for ontology matching, named MHAGINOM, is based on Multi-Head Attention Graph Isomorphism Network. This approach takes as input two OWL ontologies and performs four phases to output a set of correspondences between their concepts:

- *Preprocessing module*: This module involves reading the OWL files, creating an RDF (Resource Description Framework) graph and extracting relevant information.
- *Semantic embedding generation module.* The objective in this module is to generate semantic embeddings using the Bidirectional Encoder Representations from Transformers (BERT) model.
- *MHAGIN module*: We propose a new GNN variant that combines the expressive power of GIN and the benefits of the attention mechanism.
- *Matching module*: This module uses an MLP network to find mappings between the concepts of the input ontologies.

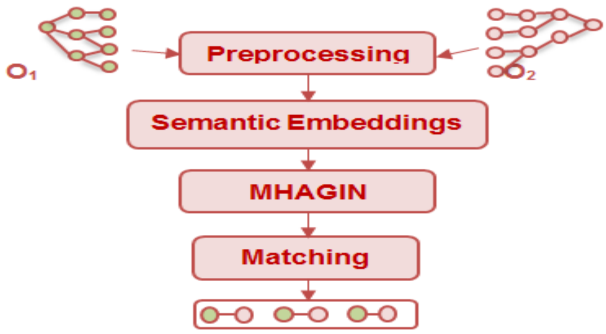

Fig. 1. MHAGINOM system architecture

4.1 Preprocessing module

The main objective of this module is to prepare raw data to be compatible as input to our MHAGIN proposed model. The process can be described in two steps:

1. *Reading OWL files and creating an RDF graph*: The RDFLib[2] library is used to read the OWL ontology files. This library also transforms the data into an RDF graph, where concepts are represented by nodes and relations by edges.
2. *Extraction of Labels and Synonyms from Ontologies*: For each concept in the input ontologies, we extract the main label (name) and its relevant synonyms. Subsequently, the extracted labels and associated synonyms are concatenated to form a comprehensive list of terms representing each concept. This step ensures that different expressions related to a concept are considered, enhancing the robustness of the generated embeddings.

4.2 Semantic embedding generation module

To generate semantic representations of ontology entities, we use the Sentence-BERT model [22]. Every expression resulting from the concatenation of labels and

[2] https://github.com/RDFLib/rdflib/blob/6.2.0/CHANGELOG.md

synonyms in the previous module is passed through the Sentence-BERT model which generates semantic embeddings.

4.3 MHAGIN module

We build a new GNN variant that brings together the expressive power of GIN and the benefits of the multi-head attention mechanism. The MHAGIN model takes as input a set of initial node features vectors $h = \{\vec{h}_1, \vec{h}_2, \ldots.\vec{h}_n\}$ such as $\vec{h}_i \epsilon R^F$ and F is the dimension of each node features vector. It performs several transformations on the set h to produce a new set $h' = \{\vec{h'}_1, \vec{h'}_2, \ldots.\vec{h'}_n\}$ where $\vec{h'}_i \epsilon\ R^{F'}$. As depicted in Fig. 2, in each layer, the process of these transformations is phased in four steps:

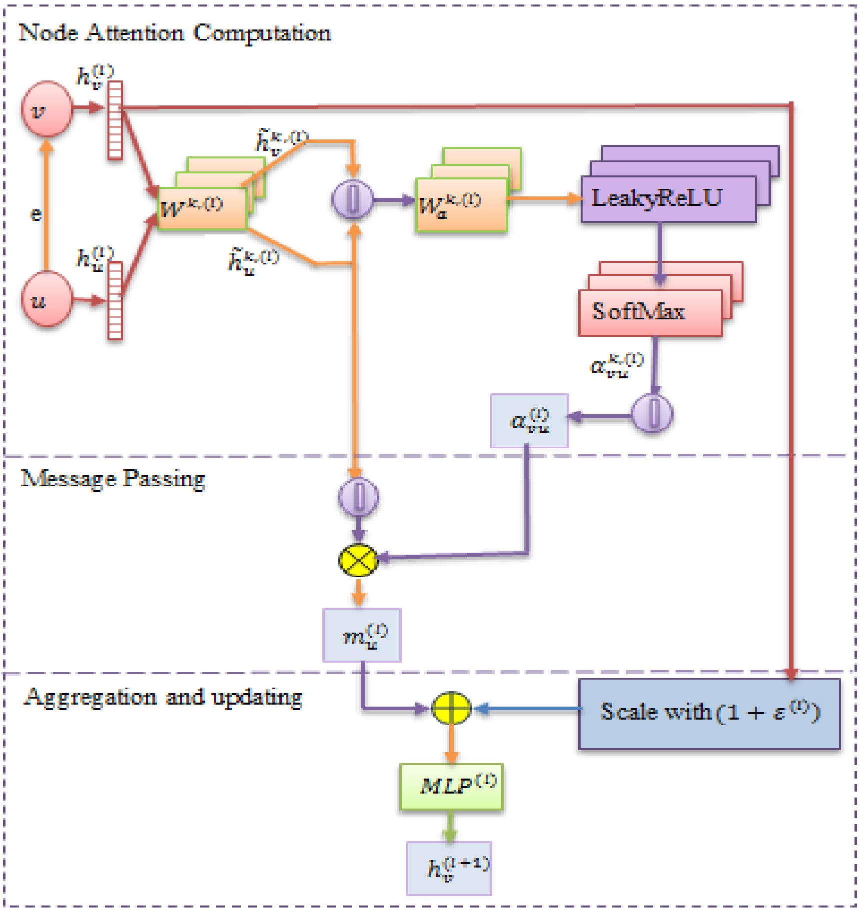

Fig. 2. Illustration of our proposed MHAGIN Model

a) Node attention computation: We utilize a multi-head attentional setup following closely the work of Veličković et al [14]. We use K independent attention mechanisms that will be concatenated. Firstly, each head k applies, on each node v, a shared linear transformation parameterized by a weight matrix $W^k \epsilon R^{F \times F'}$, such that:

$$\tilde{h}_v^{k,(l)} = W^{k,(l)} \vec{h}_v^{(l)} \tag{3}$$

Then, a shared attentional mechanism a^k: $R^{F'} \times R^{F'} \rightarrow R$ is performed:

$$e_{vu}^{k,(l)} = a^{k,(l)}(\tilde{h}_v^{k,(l)}, \tilde{h}_u^{k,(l)}) \tag{4}$$

where e_{vu}^k indicates the importance of the node u to the node v by using the head k. Then, we utilized a single-layer feed-forward neural network parameterized by a weight vector $\vec{W}_a^k$ to represent the attention mechanism a^k and we applied the LeakyReLU nonlinearity as follows [13]:

$$\alpha_{vu}^{k,(l)} = \frac{\exp\left(LeakyReLU(\vec{W}_a^{k,(l)^T}\left[\tilde{h}_v^{k,(l)} \| \tilde{h}_u^{k,(l)}\right]\right)}{\sum_{j \epsilon N_v} \exp\left(LeakyReLU(\vec{W}_a^{k,(l)^T}\left[\tilde{h}_v^{k,(l)} \| \tilde{h}_j^{k,(l)}\right]\right)} \tag{5}$$

where $.^T$ is the transposition and $\|$ is the concatenation operation.

Then, we concatenate the attention weights calculated by all the considered head:

$$\alpha_{vu}^{(l)} = \|_{k=1}^{K} \alpha_{vu}^{k,(l)} \tag{6}$$

b) **Message Passing:** This step involves exchanging information between connected nodes where each node v creates a message $m_v^{(l)}$ which will be sent to other neighbor's nodes:

$$m_v^{(l)} = \alpha_{vu}^{(l)} \|_{k=1}^{K} \tilde{h}_v^{k,(l)} \tag{7}$$

c) **Aggregation Functions:** The objective of this step is to enable each node v to integrate information from all its neighbors. To achieve this, each node v aggregates the messages received from its neighbors. Here, we opt for utilizing the SUM aggregation, which is recognized as the most expressive aggregator [15].

$${h_{agg}}_v^{(l)} = \sum_{u \in N_v} m_u^{(l)} \tag{8}$$

d) **Node Representation Update:** each node v updates its own representation by combining its current state with the aggregated information. We adopt the GIN updating representation for each node v as follows:

$$h_v^{(l+1)} = MLP^{(l)}((1 + \varepsilon^{(l)}).h_v^{(l)} + {h_{agg}}_v^{(l)}) \tag{9}$$

where ε is a learnable parameter.

4.4 Matching Module

Based on the concept representations $h'^{(l)}$ learned from our proposed MHAGIN, the matching module takes as input pairs of concept embeddings $h_c^{O_1}, h_{c'}^{O_2}$ from O_1 and O_2 and try to predict the correspondence through a calculated score [16]:

$$M\left(h_c^{O_1}, h_{c'}^{O_2}\right) = \sigma(W_2 . \gamma(W_1(h_c^{O_1} \| h_{c'}^{O_2}) + b_1) + b_2) \tag{10}$$

As the reference alignment provided in the datasets is composed by equivalence, our matching module can only predict this type of relations between concepts.

To train the matching module, we use the following loss formula:

$$\mathcal{L}^M = \sum_{(i,j) \in M^+} M\left(h_i, h_j\right) + \sum_{(i',j') \in M^-} \omega\left[\lambda - M(h_{i'}, h_{j'})\right]_+ \tag{11}$$

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

5 Experiments

We evaluate and compare the training and test performance of MHAGIN and other baselines systems.

5.1 Datasets

In He et al. [17], the authors point out limitations in the Ontology Alignment Evaluation Initiative (OAEI) tracks, particularly for ML-based systems. To address this, they proposed a new machine learning-friendly truck[3] based on Mondo[4] and UMLS[5] resources. To train and test our approach, we used Mondo datasets involving OMIM [23], ORDO [24], NCIT (National Cancer Institute Thesaurus) and DOID (Human Disease Ontology) ontologies. For every task, a reference alignment is manually constructed containing pairs of positive examples. To obtain a balanced dataset, we generated the same number of negative examples, by replacing randomly one of the concepts in the positive sample pairs.

Dataset splitting. Each dataset is divided into three sets as follow [17]: the first one corresponding to the train set represent 20%, while the second partition witch correspond to the validation set is fixed to 10%; the remaining 70% are for the test set.

[3] https://www.cs.ox.ac.uk/isg/projects/ConCur/oaei/
[4] https://mondo.monarchinitiative.org/
[5] https://www.nlm.nih.gov/research/umls/index.html

5.2 Evaluation metrics

To evaluate the performance of our proposed ontology matching approach, we adopt the same evaluation metrics given in [17].

Let m be a correspondence such that $m = (c, c')$ where $c \epsilon O_1$ and $c' \epsilon O_2$. Let $\mathcal{M}_m$be the set of negative correspondences. Because of our approach is ML-based, we use Hits@K and MRR metrics defined as follow:

$$Hits@K = \frac{\{m \in \mathcal{M}_{ref} | Rank(m) \leq K\}}{|\mathcal{M}_{ref}|} \tag{12}$$

$$MRR = \frac{\sum_{m \epsilon \mathcal{M}_{ref}} Rank(m)^{-1}}{|\mathcal{M}_{ref}|} \tag{13}$$

where $\mathcal{M}_{ref}$ denote the reference alignment and *Rank(m)* the position of *m* in the set $\mathcal{M}_m \cup \{m\}$ ordered by the score of its elements.

We also use general metrics that are Precision (P), Recall (R) and F-score (F_β).

$$P = \frac{|\mathcal{M}_{out} \cap \mathcal{M}_{ref}|}{|\mathcal{M}_{out}|} \tag{14}$$

$$R = \frac{|\mathcal{M}_{out} \cap \mathcal{M}_{ref}|}{|\mathcal{M}_{ref}|} \tag{15}$$

$$F_\beta = (1 + \beta^2).\frac{P.R}{\beta^2.P + R} \tag{16}$$

where $\mathcal{M}_{out}$ is the alignment provided by the system under evaluation and β is a weighting for Precision and Recall.

5.3 Baselines

To assess the effectiveness of our proposed approach, we conducted a comparison with the following systems:

- **LogMap.** Is a state-of-the-art rule-based ontology matching system. It employs a hybrid approach that combines both linguistic and structural techniques to discover semantic correspondences between entities in different ontologies [3].
- **AgreementMakerLight (AML).** It is a leading rule-based ontology matching system. It is an efficient tool that focuses on attribute-based matching for RDF data [4].
- **BERTMap.** a ML-based ontology matching system which adapt the BERT model on a corpus of concept labels extracted from the ontologies to be aligned [11].
- **EditSim.** According to He et al. [17], it is reasonable to consider the simple edit distance between concept labels as baseline. It represents the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another [18].

5.4 Experimental configurations

To set the hyper-parameters of our system, we used the ADAM optimizer [19] with initial learning rate 0.001. We started by varying the number of layers and epochs of the MHAGIN model according to all the metrics. Then we varied the number of MLP layers and batch size, in terms of precision, recall and F1 metrics. The following tables show the results obtained using the OMIM-ORDO dataset.

Table 1. Evaluation of MHAGIN by varying the number of epochs

# Epoch	P	R	F1	MMR	HIT@1
100	0.747	0.489	0.591	0.805	0.781
500	0.757	0.499	0.601	0.822	0.792
1000	**0.765**	**0.505**	**0.608**	**0.840**	**0.799**

Table 2. Evaluation of MHAGIN by varying the number of layers

# Layers	P	R	F1	MMR	HIT@1
1	**0.765**	0.505	0.608	0.840	0.799
2	0.757	**0.525**	**0.620**	**0.847**	**0.802**
3	0.762	0.496	0.601	0.836	0.794

Table 3. Evaluation of MLP by varying the number of layers

Number of layers	P	R	F1
1	0.426	0.526	0.471
2	0.511	0.398	0.505
4	0.572	0.451	0.505
6	**0.590**	**0.570**	**0.580**

Table 4. Evaluation of MLP by varying the batch size

# Batch size	P	R	F1
32	**0.590**	0.570	0.580
64	0.579	**0.598**	**0.588**
128	0.586	0.516	0.549
256	0.579	0.508	0.541

According to the results obtained, we will consider 2 layers for MHAGIN and 6 layers for MLP. While the best batch size obtained is equal to 64, considering the best recall and F1 score, even if the best precision was obtained with a batch size equal to 32. For the number of epochs, we'll consider the value 1000, which gave the best performance according to all the metrics.

5.5 Evaluation results

Evaluation of the different variants of the multi-head attention. We have considered the different variants (with/without the integration of synonyms), namely: the

MHAGINOM and the MHAGINOM without Synonyms (MHAGINOM-S). Table 5 shows that the MHAGINOM model outperformed the MHAGINOM-S model for both datasets, demonstrating the contribution of integrating synonyms. Accordingly, this model will be used in the rest of our evaluations.

Table 5. Contribution of synonyms

Dataset	Model	P	R	F1	MMR	HIT@1
OMIM-ORDO	MHAGINOM-S	0.757	0.545	0.634	0.871	0.814
	MHAGINOM	**0.773**	**0.560**	**0.650**	**0.887**	**0.827**
DOID-NCIT	MHAGINOM-S	0.884	0.814	0.847	0.971	0.966
	MHAGINOM	**0.889**	**0.836**	**0.862**	**0.975**	**0.968**

Contribution of multi-head attention with GIN for ontology matching. To evaluate the performance of our model, we compared it with the GIN model (without attention) and the GIN model with self-attention (SAGIN-OM). In addition, to assess the contribution of GIN, we added a comparison with the GAT graph model. According to [20], higher MMR and HIT@k scores reflect improved performance when evaluating entity matching methods. Table 6 shows the results obtained using both datasets, in terms of MMR and HIT metrics:

Table 6. Evaluation of MHAGINOM in terms of MMR and HIT

Model	OMIM-ORDO		DOID-NCIT	
	MMR	**HIT@1**	**MMR**	**HIT@1**
GAT-OM	0.827	0.791	0.893	0.873
GIN-OM	0.847	0.802	0,954	0,941
SAGIN-OM	0.869	0.811	0,969	0,959
MHAGINOM	**0.887**	**0.827**	**0.975**	**0.968**

We can notice that our approach outperformed the other methods for both datasets. Furthermore, we evaluated the ontology matching performance in terms of P, R and F1 metrics, as illustrated in Table 7.

Table 7. Evaluation of MHAGINOM in terms of P, R and F1

Model	OMIM-ORDO			DOID-NCIT		
	P	**R**	**F1**	**P**	**R**	**F1**
GAT-OM	0.728	0.503	0.595	0.867	0.765	0.813
GIN-OM	0.757	0.525	0.620	0,885	0,803	0,842
SAGIN-OM	0.764	0.537	0.631	**0,896**	0,823	0,858
MHAGINOM	**0.773**	**0.560**	**0.650**	0.889	**0.836**	**0.862**

The results obtained show that our approach outperforms the other models, achieving a better precision, recall and F1 score with OMIM-ORDO dataset. However, with DOID-NCIT dataset, our model performed better in terms of recall and F1 score but the best precision was obtained with SAGIN-OM model. We note that we have only used HIT@1, as the values obtained with a variation of K (K=5 and 10) are almost equal to 1 with all the models.

Comparison with related work. We have compared the performance of our model with the state-of-the-art using both OMIM-ORDO and DOID-NCIT datasets. The following tables illustrate the results obtained, given that we have reused the results of related work from He et al [17] since we are using the same datasets he shared:

Table 8. Comparison using the OMIM-ORDO dataset

Model	P	R	F1	MMR	HIT@1
LogMap	**0.788**	0.501	0.612	0.805	0.744
AML	0.702	0.517	0.596	NA	NA
BERTMap	0.762	0.548	0.637	0.877	0.823
EditSim	0.781	0.507	0.615	0.777	0.727
MHAGINOM	0.773	**0.560**	**0.650**	**0.887**	**0.827**

Table 9. Comparison using the DOID-NCIT dataset

Model	P	R	F1	MMR	HIT@1
LogMap	**0.896**	0.661	0.761	0.559	0.363
AML	0.841	0.770	0.804	NA	NA
BERTMap	0.823	**0.887**	0.854	0.968	0.955
EditSim	0.889	0.771	0.826	0.903	0.883
MHAGINOM	0.889	0.836	**0.862**	**0.975**	**0.968**

The comparison results show that with the OMIM-ORDO dataset, our model outperformed the state-of-the-art models in terms of recall, F1 score, MMR and HIT@1 metrics. For precision, the best value obtained was 0.788 with LogMap against a value of 0.773 with our model. However, with the DOID-NCIT dataset our model outperformed the other models in terms of F1 score, MMR and HIT@1 metrics. For precision, the best value obtained was 0.896 with LogMap against a value of 0.889 with our model. Similarly, for recall, the best value obtained was 0.887 with BERTMap against a value of 0.836 with our model.

5.6 Discussion and analysis

Experiments carried out on two different datasets demonstrate that the ontology matching based on multi-head with attention mechanism and using GIN outperformed the model based on self-attention GIN as well as the GIN or other types of knowledge graph models such as the GAT model.

Moreover, the results obtained prove that our model outperformed the state-of-the-art models in terms of F1 score, MMR and HIT@1 metrics. We can notice that the most precise model is LogMap, but our model yielded a better recall and F1 score than this model for both datasets. On the other hand, although BERTMap provided a better recall with the DOID-NCIT dataset, our model was more precise with both datasets and also yielded a better recall and F1 score than this model. Therefore, these evaluations prove that our model is the most effective for ontology matching.

However, the major shortcomings of our contribution lies in the fact that we have considered only one type of mapping, namely the equivalence, and one type of relation, the "is-a" relation. It would be interesting to extend our approach by considering other types of mapping, such as the subsumption [21], and other types of semantic relations, such as "part-of", "has-a" and so on.

6 Conclusion

We proposed in this article a multi-head attention graph isomorphism network for ontology matching. Our method consists of three steps: ontology pre-processing, generating embeddings with BERT; the MHAGIN module, proposing a novel GNN variant that combines the expressive power of GIN and the benefits of the attention mechanism; and the matching module, which uses an MLP network to find mappings between the concepts of the input ontologies. The experiment conducted on two different datasets showed that our approach outperformed existing state-of-the-art methods, proving the contribution of multi-head GIN and attention mechanism along with the integration of synonyms.

As future work, we will consider different types of relations not only the “is-a” relation and other types of mapping such as the subsumption. Furthermore, we plan to deploy our solution in a real-world environment, developing applications in a variety of contexts including e-learning, e-tourism and bioinformatics.

References

1. Gruber, T. R. A translation approach to portable ontology specifications, Knowledge Acquisition 5 (2) (1993) 199–220, https://www. sciencedirect.com/science/article/pii/S1042814383710083.
2. Kalfoglou, Y., Schorlemmer, M. Ontology mapping: the state of the art, The Knowledge Engineering Review 18 (1) (2003) 1–31. doi:10.1017/S0269888903000651.
3. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I. Large-scale interactive ontology matching: Algorithms and implementation, in: ECAI, Vol. 242, (2012), pp. 444–449, https://doi.org/10.3233/978-1-61499-098-7-444.
4. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., Couto, F.M. The AgreementMakerLight ontology matching system, in: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer Berlin Heidelberg, (2013), pp. 527–541, https://doi.org/10. 1007/978-3-642-41030-7_38.

5. Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J. Ontology matching with word embeddings, in Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer, Cham, (2014), pp.34–45. https://doi.org/10.1007/978-3-319-12277-9_4
6. Wang, L., Bhagavatula, C., Neumann, M., Lo, K., Wilhelm, C., Ammar, W. 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.
7. Iyer, V., Agarwal, A., Kumar, H. Veealign: a supervised deep learning approach to ontology alignment, in: Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, (2020), Vol. 2788 of CEUR Workshop Proceedings, CEUR-WS.org, 2020, pp. 216-224, http://ceur-ws.org/Vol-2788/oaei20_paper13.pdf.
8. Wang P., Hu Y. Matching Biomedical Ontologies via a Hybrid Graph Attention Network. Front. Genet. 13:893409 (2022), doi: 10.3389/fgene.2022.893409.
9. Xu, K., Hu, W., Leskovec, J., Jegelka, S. How powerful are graph neural networks?, In International Conference on Learning Representations (ICLR), (2028). arXiv 2018:00826.1810.
10. Bento, A., Zouaq, A., Gagnon, M. Ontology Matching Using Convolutional Neural Networks. In Proceedings of the Twelfth Language Resources and Evaluation Conference, (2020), pp. 5648–5653, Marseille, France. European Language Resources Association.
11. He, Y., Chen, J., Antonyrajah, D., Horrocks, I. BERTMap: a BERT-based ontology alignment system. Proceedings of the AAAI Conference on Artificial Intelligence, ojs.aaai.org, (2022).
12. Wang, P., Hu, Y. Matching Biomedical Ontologies via a Hybrid Graph Attention Network. Front Genet. (2022), 13:893409. doi: 10.3389/fgene.2022.893409. PMID: 35938027; PMCID: PMC9354052.
13. Hu, Y., Tang, Y., Huang, H., He, L. A Graph Isomorphism Network with Weighted Multiple Aggregators for Speech Emotion Recognition. Proc. Interspeech 2022, 4705-4709, (2022). doi: 10.21437/Interspeech.2022-637
14. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. Graph attention networks. (2017). *arXiv preprint arXiv:1710.10903*.
15. Ying, J., Z., Leskovec, J. Design space for graph neural networks. Advances in Neural Information Processing Systems 33 (2020), 17009–17021.
16. Hao, J., Lei, C., Efthymiou, V., Quamar, A., Özcan, F., Sun, Y., Wang, W. 2021. MEDTO: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, (2021), Virtual Event, Singapore. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467138
17. He, Y., Chen, J., Dong, H., Jiménez-Ruiz, E., Hadian, A., Horrocks, I. Machine learning-friendly biomedical datasets for equivalence and subsumption ontology matching, arXiv preprint arXiv:2205.03447 (2022).
18. Lê Bach T. Building a Multi-Perspective Semantic Web. PhD thesis in Computer Science. École des Mines de Nice at Sophia Antipolis, (2006).
19. Kingma, D., Ba, J., Adam. A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR'15), (2015).
20. Cai, W., Ma, W., Zhan, J., Jiang, Y. Entity Alignment with Reliable Path Reasoning and Relation-Aware Heterogeneous Graph Transformer. International Joint Conference on Artificial Intelligence (2022).

21. Chen, J., He, Y., Geng, Y., Jimenez-Ruiz, E., Dong, H., Horrocks, I. Contextual semantic embeddings for ontology subsumption prediction. *World Wide Web*, (2023), p. 1-23.
22. Reimers, N., & Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In Proceedings of Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. (2019). arXiv preprint arXiv:1908.10084.
23. Amberger, J.S., Bocchini, C.A., Schiettecatte, F., Scott, A.F., Hamosh, A.: OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. Nucleic acids research 43(D1), D789–D798 (2015)
24. Vasant, D., Chanas, L., Malone, J., Hanauer, M., Olry, A., Jupp, S., Robinson, P.N., Parkinson, H., Rath, A.: ORDO: an ontology connecting rare disease, epidemiology and genetic data. In: Proceedings of ISMB. vol. 30 (2014).