



A GNN-Based Framework for Ontology Matching

Samira Oulefki^(✉) and Lamia Berkani

USTHB, Bab Ezzouar, 16111 Algiers, Algeria
`{s.oulefki,lberkani}@usthb.dz`

Abstract. Ontology Matching (OM) is a critical task within the Semantic Web, addressing the challenge of semantic heterogeneity by identifying correspondences between entities from different ontologies. Traditional approaches often depend on predefined rules and external lexicons, while machine learning (ML) methods, including Deep Learning (DL) and Graph Neural Networks (GNNs), have gained traction due to their ability to generate expressive entity representations. However, existing GNN-based OM methods tend to focus predominantly on Graph Convolutional Networks (GCNs), overlooking the potential of other GNN architectures. To address this limitation, we introduce GNN-Match, a versatile framework that incorporates multiple GNN variants—Graph Transformer Networks (GTNs), Graph Attention Networks (GATs), Graph Isomorphism Networks (GINs), and GCNs. This framework allows users to select the most suitable GNN model to enhance semantic embeddings generated by a BERT-based encoder and capture structural features. Users can then aggregate these features using methods such as gated networks and concatenation, effectively capturing both semantic and structural information to optimize concept mapping. Extensive experiments across diverse datasets demonstrate that GNN variants consistently outperform non-GNN systems and offer new insights into the role of various GNN architectures in OM tasks.

Keywords: Ontology Matching · Graph Neural Network · Graph Transformer

1 Introduction

Ontologies have become indispensable tools for enabling effective communication and knowledge sharing between humans and machines. By providing a shared vocabulary and structure for specific domains, ontologies are widely used in databases, Large Language Models (LLMs), and recommendation systems. However, as the number of ontologies grows across various domains, semantic heterogeneity becomes a significant challenge, complicating data integration and interoperability. Ontology matching (OM) addresses this challenge by identifying correspondences between entities from different ontologies, enabling seamless integration across diverse domains [1].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
C. Ordonez et al. (Eds.): MEDI 2024, LNCS 15590, pp. 183–197, 2025.

https://doi.org/10.1007/978-3-031-87719-3_14

OM involves determining semantic relationships between concepts, relations, and instances from different ontologies, ultimately producing a set of mappings (alignment). Formally, given two ontologies O_1 and O_2 , each comprising sets of concepts (C_i), relations (R_i), axioms (A_i), and instances (I_i), OM can be defined as a function f , where $f(O_1, O_2) \rightarrow M$, with M representing a set of mappings $\{m_k\}_{k=1}^L$. Each mapping $m_k = \langle v, v', r, n \rangle$ consists of entities $v \in O_1$ and $v' \in O_2$, where r denotes the predicted semantic relation between these entities, and n represents the confidence level of this relation [1]. Figure 1 illustrates an example of mappings between two scientific research ontologies. For instance, the mapping m_1 can be formalized as:

$$m_1 = \langle \text{"Conference"}, \text{"Gathering"}, \text{"="}, 0.95 \rangle$$

where “Conference” $\in O_1$ and “Gathering” $\in O_2$ are the matched concepts, “=” is the equivalence relation, r , holding between “Conference” and “Gathering” and 0.95 is the mapping score, n .

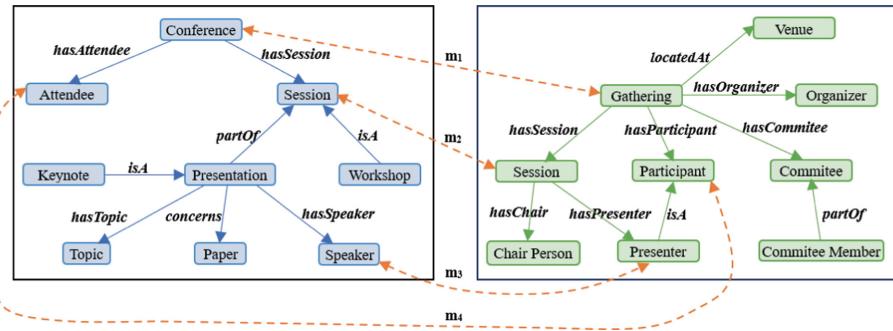


Fig. 1. An example of mappings between scientific research ontologies

Over the years, OM approaches have evolved, generally falling into two categories: traditional methods and learn approaches. Traditional methods, such as LogMap [2] and AgreementMakerLight (AML) [3], have laid a strong foundation by using lexical indexing, external references, and logical reasoning to match entities. However, these methods frequently struggle to capture complex semantic contexts.

The rise of deep learning (DL) has revolutionized OM by introducing sophisticated techniques for modeling and comparing entities within ontologies. Early work, such as that by Zhang et al. [4], utilized Word2Vec [5] embeddings to represent ontological concepts. More recent advancements, exemplified by BERTMap [6], employ pre-trained models, like BERT [7], to encode concept names and synonyms, thereby providing a deeper semantic understanding.

Recently, graph representation learning has emerged as a powerful technique in ML-based OM tasks. By representing ontologies as graphs and utilizing Graph

Neural Networks (GNNs), these approaches can aggregate information from neighboring nodes, generating robust and context-aware vector representations of entities. A prevalent trend in GNN-based OM systems has been the reliance on Graph Convolutional Networks (GCNs) [8], which, while effective, has limited the exploration of alternative GNN architectures [9, 10]. This reliance points to an important gap in the diversity of approaches for leveraging GNNs in OM.

Addressing this gap, Oulefki et al. [11] introduced the Multi-Head Attention Graph Isomorphism Network for OM (MHAGINOM), which incorporates multi-head attention mechanisms within the Graph Isomorphism Network (GIN) [10]. GIN is renowned for its high expressiveness in distinguishing graph structures, and MHAGINOM enhances this capability by employing attention heads [12] to prioritize the most relevant concepts during the aggregation process. The results from MHAGINOM demonstrate its superior performance over prevailing GCN-based models, underscoring the potential of alternative GNN variants in advancing OM.

Motivated by the promising results of alternative GNN variants like MHAGINOM, we propose a comprehensive framework for OM, GNN-Match, which incorporates multiple GNN architectures, specifically Graph Transformers (GT) [13], Graph Attention Networks (GAT) [9], Graph Isomorphism Networks (GIN), and Graph Convolutional Networks (GCN). By integrating these diverse GNN models, our framework provides insights into the performance of different GNN architectures across various OM scenarios. Extensive experiments on multiple datasets will elucidate the strengths and limitations of each GNN variant, offering a clearer direction for future research in OM.

The structure of this paper is as follows: Sect. 2 reviews existing work on OM, with an emphasis on ML, DL, and GNN techniques, while highlighting the lack of exploration into alternative GNN architectures. Section 3 outlines our proposed framework. Section 4 describes the experimental setup and presents the results, providing a detailed performance analysis of each GNN variant across various OM tasks. Finally, Sect. 5 concludes this work by summarizing key findings and offering suggestions for future research

2 Related Work

OM methods are typically divided into traditional and learned approaches. Traditional systems, such as LogMap and AML, have laid the groundwork for OM. LogMap leverages lexical indexing to create *anchor correspondences* between ontologies, enabling large-scale matching with over 100,000 concepts through a blend of linguistic and logical reasoning techniques. In contrast, AML employs an external reference ontology to mediate and execute cross-searches of bridge mappings, integrating diverse strategies including linguistic, structural, and instance-based techniques. Despite their foundational contributions, both LogMap and AML encounter limitations in effectively capturing complex semantic contexts and textual nuances, which constrains their performance across varied and intricate scenarios.

To address these limitations, learned methods have emerged as promising alternatives. These methods can be categorized into ML-based, DL-based, and GNN-based approaches. ML-based methods extract features from entities to train various algorithms such as k-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees, and AdaBoost [14]. For instance, GLUE [15] utilizes diverse learning strategies to handle different types of information, though its performance depends on the availability of sufficient data instances. VersaMatch [16], a weakly-supervised system, generates detailed alignment profiles by integrating properties, neighbors, and statistical features, and leverages NLP techniques and knowledge from Wikidata, alongside MinHash Locality-Sensitive Hashing (LSH) for efficient pair reduction.

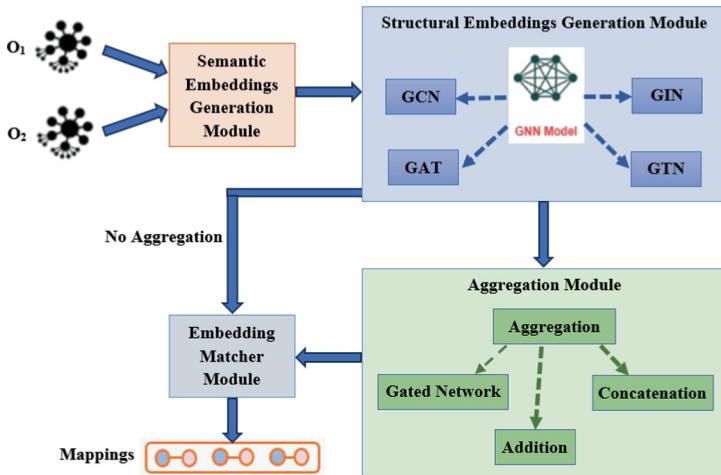


Fig. 2. The overall architecture of GNN-Match

DL-based approaches enhance matching accuracy through neural network architectures. ERSOM [17], for example, uses stacked auto-encoders to embed detailed entity descriptions and propagate similarities, while DeepAlignment [18] extends this approach by incorporating semantic nuances such as synonyms and antonyms.

In the realm of GNN-based methods, the graph structure of ontologies is utilized to capture complex relationships between entities. GCN-Align [19] applies GCNs to capture both structural and attribute information of entities, generating high-quality embedding vectors for similarity evaluation. Recently, MHAGI-NOM has integrated attention mechanisms with GINs, enhancing performance by focusing on relevant nodes during aggregation. Additionally, in the biomedical domain, methods such as DAEOM [20] and MEDTO [21] have used attention mechanisms and hyperbolic embeddings to capture complex hierarchical and relational structures.

3 GNN-Match Framework

As depicted in Fig. 2, our proposed OM framework, GNN-Match, is designed to process two OWL (Web Ontology Language) ontologies, O_1 and O_2 . The framework consists of four distinct modules: Semantic Embeddings Generation, Structural Embeddings Generation, Aggregation, and Embedding Matcher. Each module plays a specialized role in the OM process, contributing to the generation and integration of semantic and structural features to optimize the identification of mappings between concepts from the input ontologies.

3.1 Semantic Embeddings Generation Module

To effectively model concept features, we have utilized the pre-trained Sentence-BERT language model to encode concept labels and synonyms extracted from the input ontologies. For biomedical contexts, we employ BioBERT [22], which is pre-trained on an extensive corpus of biomedical literature, including PubMed abstracts and clinical notes. This specialized training equips BioBERT to capture the specific terminology, nuances, and contextual details of the biomedical domain. Consequently, for each ontology O_i , we derive a set of semantic embeddings for its concepts, denoted as $E_{sm}^{(O_i)}$. Here, $E_{sm}^{(O_i)} = \{h_{smv}^{(O_i)}\}_{v=1}^N$, where $h_{smv}^{(O_i)}$ represents the semantic embedding vector of each concept v , and N denotes the total number of concepts in the ontology O_i .

3.2 Structural Embeddings Generation Module

In our OM framework, structural embeddings can be generated using various GNN variants, allowing users to tailor the model selection to their specific requirements. Each GNN architecture brings unique advantages in capturing the structural intricacies of ontologies, a critical factor for achieving accurate and meaningful matches. In this graph-based representation, nodes correspond to concepts, and edges represent relationships between them.

Graph Convolutional Networks (GCNs). GCNs operate by aggregating node features from neighboring nodes through a series of convolutional layers, enabling them to capture local structural information and propagate it throughout the graph. The update rule at layer $(l+1)$ is given by the following formula:

$$h_v^{(l+1)} = \sigma \left(W^{(l)} h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} W^{(l)} h_u^{(l)} \right) \quad (1)$$

where:

- $h_i^{(l+1)}$ is the updated feature vector for node v at layer $l+1$,
- $h_v^{(l)}$ is the feature vector of node v at layer l ,
- $W^{(l)}$ is the learnable weight matrix at layer l ,

- $\mathcal{N}(v)$ is the set of neighbors of node v ,
- σ is a non-linear activation function, such as ReLU,
- $\sum_{u \in \mathcal{N}(v)}$ denotes the aggregation of features from the neighboring nodes.

Graph Attention Networks (GATs). GATs improve upon GCNs by incorporating an attention mechanism that assigns different importance to neighboring nodes. The update formula for GATs is as follows:

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)} \right) \quad (2)$$

where $W^{(l)}$ is the learnable weight matrix at layer l , and $\alpha_{ij}^{(l)}$ is the attention coefficient between nodes i and j , calculated as:

$$\alpha_{ij}^{(l)} = \frac{\exp(\text{LeakyReLU}(fa^T [W^{(l)} h_i^{(l)}, W^{(l)} h_j^{(l)}]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T [W^{(l)} h_i^{(l)}, W^{(l)} h_k^{(l)}]))} \quad (3)$$

where a^T is the learnable attention vector. GATs excel in settings where the relationships between nodes vary in importance, allowing them to weigh neighbors differently.

Graph Isomorphism Networks (GINs). GINs are highly expressive and are capable of distinguishing between different graph structures. The core of GIN is the aggregation function, which is typically a sum followed by a learnable MLP (Multilayer Perceptron):

$$h_i^{(l+1)} = \text{MLP}^{(l)} \left((1 + \epsilon) h_i^{(l)} + \sum_{j \in \mathcal{N}(i)} h_j^{(l)} \right) \quad (4)$$

where ϵ is a learnable scalar. This architecture allows GINs to preserve the uniqueness of graph structures, making them suitable for OM tasks requiring precise structural distinctions.

Graph Transformer Networks (GTNs). GTNs extend the GNN framework by incorporating transformer-like mechanisms to model long-range dependencies and complex patterns within the graph. The layer update rule for GTNs is given by:

$$h_i^{(l+1)} = \text{MultiHeadSelfAttention}(H^{(l)}) \quad (5)$$

where $H^{(l)} = [h_1^{(l)}, h_2^{(l)}, \dots, h_N^{(l)}]$ is the matrix of node feature vectors at layer l , and N is the total number of nodes. The multi-head self-attention mechanism is defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (6)$$

where:

- $Q = W_Q H^{(l)}$ is the query matrix,
- $K = W_K H^{(l)}$ is the key matrix,
- $V = W_V H^{(l)}$ is the value matrix,
- W_Q , W_K , and W_V are learned weight matrices,
- d_k is the dimensionality of the key vectors.

Here, the multi-head self-attention mechanism computes attention scores across all nodes, enabling the model to aggregate information from distant nodes. This flexibility makes GTNs particularly effective for handling large ontologies with intricate structural relationships.

Output Layer. By stacking L GNN layers, we obtain the structural representations of concepts within the entire ontology O_i . Specifically, the GNN model generates a unified set of structural embeddings for the concepts, denoted as $E_{str}^{O_i}$, where $E_{str}^{O_i} = \{h_{str_v}^{O_i}\}_{v=1}^N$ and $h_{str_v}^{O_i}$ represents the final structural embedding of each concept v .

3.3 Aggregation Module

The Aggregation module is pivotal in seamlessly combining semantic and structural embeddings to create a unified representation. It offers three distinct strategies for merging the embeddings: *Gated Network* [23], which dynamically adjusts the influence of each embedding; *Concatenation*, which appends embeddings together; *Addition*, which fuses embeddings element-wise; and an option where *no aggregation* is applied, allowing embeddings to pass through unchanged for downstream processing.

Gated Network. The *Gated Network* is a two-step mechanism that selectively merges semantic and structural embeddings. By learning the relative importance of each embedding, the network dynamically adjusts the weighting between them, allowing the system to prioritize the most relevant information.

Calculation of Gating Values: For each concept $v \in O_i$, given its semantic embedding, $h_{sm_v}^{O_i}$, and structural embedding, $h_{str_v}^{O_i}$, a linear transformation followed by a sigmoid activation is applied to compute the gating value as follows:

$$g_{sm_v} = \sigma(W_g h_{sm_v}^{O_i}) \quad (7)$$

where W_g is a weight matrix, and σ is the sigmoid activation function.

Embedding Combination: The embeddings are then combined using the gating values:

$$h_{g_v}^{O_i} = g_{sm_v} \odot h_{sm_v}^{O_i} + (1 - g_{sm_v}) \odot h_{str_v}^{O_i} \quad (8)$$

where \odot represents the element-wise product.

Concatenation. The *Concatenation* method merges embeddings by appending them into a unified vector. For each concept $v \in O_i$, the semantic embedding and structural embedding are combined as follows:

$$h_{g_v}^{O_i} = [h_{sm_v}^{O_i} \parallel h_{str_v}^{O_i}] \quad (9)$$

where, \parallel represents concatenation.

Addition. The *Addition* method combines the semantic and structural embeddings of each concept $v \in O_i$ by performing an element-wise sum :

$$h_{g_v}^{O_i} = [h_{sm_v}^{O_i} + h_{str_v}^{O_i}] \quad (10)$$

No Aggregation. In this option, the embeddings generated by the *Structural Embedding Generation* module are directly passed as inputs to the next module without any aggregation.

3.4 Embedding Matcher Module

The final concept embeddings of the ontologies O_1 and O_2 are compared using cosine similarity to determine potential mappings between concepts. The cosine similarity between two embeddings, $h_{g_v}^{O_1}$ from O_1 and $h_{g_{v'}}^{O_2}$ from O_2 , is computed as follows:

$$\text{sim}_{cos} \left(h_{g_v}^{O_1}, h_{g_{v'}}^{O_2} \right) = \frac{\sum_{i=1}^{d_c} h_{g_{v_i}}^{O_1} \cdot h_{g_{v'_i}}^{O_2}}{\sqrt{\sum_{i=1}^{d_c} \left(h_{g_{v_i}}^{O_1} \right)^2} \cdot \sqrt{\sum_{i=1}^{d_c} \left(h_{g_{v'_i}}^{O_2} \right)^2}} \quad (11)$$

where d_c is the dimensionality of the concept embeddings.

A valid mapping is established when the cosine similarity value, $\text{sim}_{cos} \left(h_{g_v}^{c,O_1}, h_{g_{v'}}^{c,O_2} \right)$, exceeds a predefined threshold, indicating sufficient similarity between the corresponding concept embeddings from both ontologies.

4 Experiments

In this section, we present a comprehensive overview of our experimental process, covering the datasets, evaluation metrics, experimental setup, baseline systems, and a detailed analysis of the results. Our proposed framework was implemented

using Python and executed on Google Colab. The experiments were run in a computational environment equipped with an Intel Xeon 2vCPU @ 2.3 GHz processor, an Nvidia Tesla K4 GPU with 16 GB of memory, and 13 GB of system RAM. This configuration ensured efficient handling of the computational demands during experimentation.

4.1 Datasets

Table 1. Ontology Pair Statistics

Ontology Pair	#Concepts	#Relations	#Reference (=)
SNOMED-NCIT (Neoplas)	11,271–13,956	1,280–30,881	3,804
AMA-(NCI-A)	2,744–3,304	4,500–5,500	1,544

To assess the performance of the different GNN architectures included in our framework, we utilized two widely recognized datasets from the Ontology Alignment Evaluation Initiative (OAEI)¹, both sourced from the biomedical domain and each posing distinct challenges: the {SNOMED-CT [24], NCIT (Neoplas) [25]} dataset, as well as the Anatomy task, which involves aligning the Adult Mouse Anatomy Dictionary [26] with the NCI Thesaurus on human anatomy. These datasets pose considerable challenges due to their large scale, complex terminologies, and hierarchical relationships, making them ideal for evaluating OM systems. Table 1 provides a summary of the number of concepts, relations, and the size of the reference alignments for these datasets.

For the evaluation, the provided reference alignments were split into training, validation, and test sets, serving as positive samples. To create negative samples, we generated Q negative correspondences for each positive match by randomly replacing one concept from the positive pairs with another concept from the corresponding ontology, ensuring that the evaluation remained balanced and rigorous.

4.2 Evaluation Metrics

Our framework is assessed using precision, recall, and $F1$ score. Precision (P) quantifies the proportion of true positives among all predicted positives, reflecting the accuracy of the predicted correspondences. Recall (R) measures the proportion of true positives identified from the total number of actual positives, highlighting the system’s capability to detect all relevant matches. The $F1$ score combines precision and recall into a single metric by computing their harmonic mean, offering a balanced measure of performance.

¹ <http://oaei.ontologymatching.org/>.

4.3 Experimental Settings

To optimize performance and manage memory, the embedding dimension was set to 384. During dataset preparation, the number of negative examples (Q) was adjusted based on the dataset size: Q was set to 100 for the SNOMED-NCIT (Neoplas) task and 20 for the Anatomy task. We used the ADAM optimizer [27] with a learning rate of 0.001 to ensure effective convergence. For the aggregation module, the gated network was chosen due to its superior performance. The threshold for the embedding matcher was empirically set to 0.5, based on observed performance across various ontologies.

4.4 Baselines

For benchmarking, we selected several state-of-the-art OM systems as baselines:

- **AML** [3]: A leading system that integrates lexical, mediating, and word-based string similarity matchers to produce a comprehensive alignment between input ontologies.
- **LogMap** [2]: A state-of-the-art system that utilizes a hybrid approach, combining linguistic and structural techniques to identify semantic correspondences.
- **LogMapBio** [28]: An extension of LogMap that leverages BioPortal [29] as a source of mediating ontologies, moving beyond a reliance on a few preselected ontologies.
- **Matcha-DL** [30]: A DL-based framework that employs supervised learning techniques to refine decision boundaries for mappings.
- **VersaMatch** [16]: A system that uses alignment profiles, MinHash-based blocking, and ensemble methods to classify pairs of concepts.

4.5 Experimental Results

Table 2 summarizes the performance results of several OM systems and the different GNN architectures incorporated in GNN-Match, evaluated on the Anatomy and Neoplas datasets. The effectiveness of each system is quantified through key metrics, including precision, recall, and F1 score, providing a detailed insight into their capabilities in aligning complex ontological structures.

Results and Analysis on the Anatomy Dataset. As depicted in Table 2 and Fig. 3, GNN-Match with GTN achieves the highest $F1$ score of 0.943 on the Anatomy dataset, indicating the most effective balance between precision and recall among the evaluated systems. VersaMatch leads in precision with a score of 0.960, reflecting its high accuracy in establishing correct correspondences, though it has a relatively lower recall of 0.880. In contrast, GNN-Match with GAT exhibits the highest recall at 0.935, indicating its effectiveness in identifying

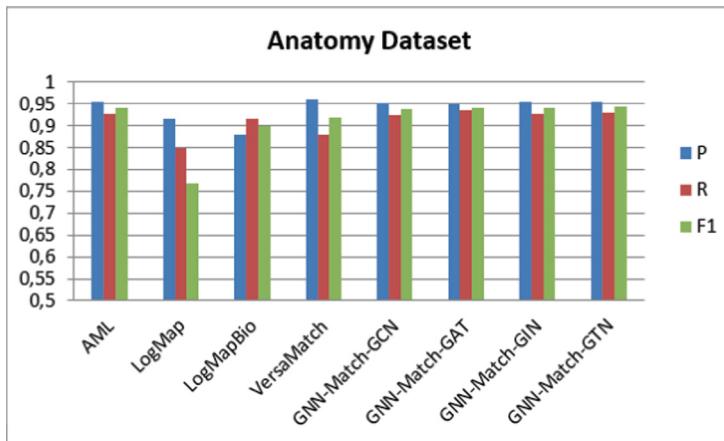


Fig. 3. Performance of models on the Anatomy dataset

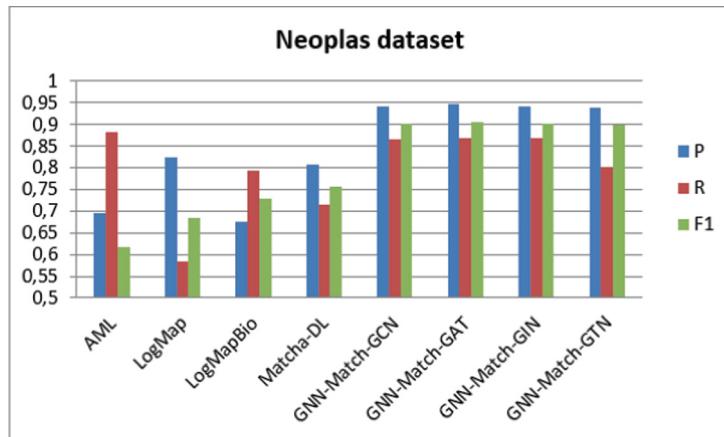


Fig. 4. Performance of models on the Neoplas dataset

a wide range of relevant matches. Additionally, AML and GNN-Match with GIN also perform commendably, with *F1* scores of 0.941 and 0.940, respectively.

Compared to GNN-Match with GAT, which has the second-highest *F1* score, GNN-Match with GTN shows a 0.11% improvement in *F1* score and a 0.74% increase in precision. However, its recall is slightly lower by 3.75%.

Results and Analysis on the Neoplas Dataset. The results depicted in Table 2 and Fig. 4 show that GNN-Match with GAT is the best-performing system on the Neoplas dataset, achieving an *F1* score of 0.906, a precision of 0.946, and a recall of 0.870.

Table 2. Comparison of GNN variants with Baseline Systems on Anatomy and Neoplas Datasets.

Model	Anatomy			Neoplas		
	F1	P	R	F1	P	R
AML [3]	0.941	<u>0.956</u>	0.927	0.616	0.696	0.552
LogMap [2]	0.767	0.917	0.848	0.683	0.823	0.583
LogMapBio [28]	0.898	0.880	0.916	0.729	0.675	0.793
Matcha-DL [30]	—	—	—	0.757	0.806	0.714
VersaMatch [16]	0.920	0.960	0.880	—	—	—
GNN-Match with GCN	0.938	0.953	0.924	0.902	<u>0.941</u>	0.867
GNN-Match with GAT	<u>0.942</u>	0.949	0.935	0.906	0.946	0.870
GNN-Match with GIN	0.940	0.954	0.926	<u>0.903</u>	0.940	<u>0.868</u>
GNN-Match with GTN	0.943	0.956	0.929	0.900	0.938	0.862

Compared to GNN-Match with GIN, which ranks second in *F1* score, GNN-Match with GAT demonstrates a 0.33% improvement in *F1* score and a 0.64% increase in precision. Additionally, GNN-Match with GAT achieves a marginally higher recall, exceeding GNN-Match with GIN by 0.23%.

4.6 Discussion

This study shows that GNN variants consistently outperform non-GNN systems across all considered datasets, highlighting the superior capabilities of GNN-based models in OM tasks. By exploiting the structural characteristics of ontologies, GNNs offer enhanced contextualization and deeper semantic understanding, resulting in more precise concept mappings. While non-GNN systems demonstrate strong performance in specific metrics, such as precision, they tend to fall short in overall adaptability and performance.

The variation in performance among GNN variants also highlights the influence of dataset characteristics. In the Anatomy dataset, GTN stands out by leveraging the highly structured and hierarchical nature of anatomical data. On the other hand, in the Neoplas dataset, GAT performs best, likely due to its ability to handle more complex and intertwined relationships between concepts, typical of biomedical ontologies. This variation suggests that GNN performance is closely tied to the structural complexity of the dataset, indicating that different GNN architectures may be more suited to specific types of ontologies.

However, our framework does not include emerging GNN models, such as hybrid architectures that integrate different types of graph neural networks. Additionally, it lacks user-customizable parameters, which may limit optimization for specific datasets or tasks. Addressing these limitations could enhance the framework's flexibility and effectiveness.

5 Conclusion

In this paper, we introduce a GNN-based framework, GNN-Match, that assesses various GNN variants to address the challenges of OM. Our experiments demonstrate that GNN-based approaches outperform non-GNN systems, proving their superior adaptability and capability in handling complex relationships within ontologies. The performance of different GNN variants varies depending on the dataset characteristics; for instance, GTN excels in highly structured datasets like Anatomy, while GAT performs best in more complex, heterogeneous environments such as the Neoplas dataset. This variation in performance underscores the importance of selecting the appropriate GNN architecture based on the structure of the target ontologies. Ultimately, our framework not only fills the gap in exploring diverse GNN architectures for OM, but also sets the stage for future research focused on fine-tuning these models to further enhance OM effectiveness.

Moving forward, future research should aim to expand the GNN-Match framework by incorporating a broader range of GNN variants and hybrid models to boost performance across various ontological structures. Additionally, expanding evaluations to include a diverse set of datasets and integrating real-world feedback will be crucial for validating and refining the framework's effectiveness.

References

1. Euzenat, J., Shvaiko P.: *Ontology Matching*. Springer-Verlag, Heidelberg (2013)
2. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: logic-based and scalable ontology matching. In: Aroyo, L., et al. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 273–288. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_18
3. Slaiimi, F., Sellami, S., Boucelma, O., Ben Hassine, A.: Flexible matchmaking for RESTful web services. In: Meersman, R., et al. (eds.) OTM 2013. LNCS, vol. 8185, pp. 542–554. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41030-7_39
4. Van Woensel, W., Al Haider, N., Ahmad, A., Abidi, S.: A cross-platform benchmark framework for mobile semantic web reasoning engines. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 389–408. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_25
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Neural Information Processing Systems, pp. 3111–3119 (2013)
6. He, S., Chen, J., Jiménez-Ruiz, E., Antony, B., Horrocks, I.: BERTMap: a BERT-based ontology alignment system. In: IJCAI 2022, pp. 3675–3681 (2022). <https://doi.org/10.24963/ijcai.2022/509>
7. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
8. Kipf, T. N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)

9. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (ICLR) (2018)
10. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (ICLR) (2019)
11. Oulefki, S., Berkani, L., Boudjenah, N., Kenai, I. E., Mokhtari, A.: Ontology matching using multi-head attention graph isomorphism network. In: International Conference on Model and Data Engineering, pp. 200–213 (2023)
12. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
13. Ying, Z., et al.: Transformers for graphs: a survey (2021)
14. Shadgara, B., Nejhadi, A.H., Osareh, A.: Ontology alignment using machine learning techniques. *Int. J. Comput. Sci. Inf. Technol.* **3**(2), 139–150 (2011)
15. Doan, A., Madhavan, J., Domingos, P. M., Halevy, A. Y.: Ontology matching: a machine learning approach. In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*, pp. 385–404. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24750-0_19
16. Fürst, J., Argerich, M.F., Cheng, B.: VersaMatch: ontology matching with weak supervision. *VLDB Endow.* **16**(6), 1305–1318 (2023)
17. Xiang, C., Jiang, T., Chang, B., Sui, Z.: ERSOM: a structural ontology matching approach using automatically learned entity representation. In: Conference on Empirical Methods in Natural Language Processing, pp. 2419–2429 (2015)
18. Kolyvakis, P., Kalousis, A., Kiritsis, D.: Deepalignment: unsupervised ontology matching with refined word vectors. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2018)
19. Wang, Z., Lv, Q., Lan, X., Zhang, Y.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: Conference on Empirical Methods in Natural Language Processing, pp. 349–357 (2018)
20. Wu, J., Lv, J., Guo, H., Ma, S.: DAEOM: a deep attentional embedding approach for biomedical ontology matching. *Appl. Sci.* **10**(21) (2020)
21. Hao, J., Lei, C., Efthymiou, V., Quamar, A., Özcan, F., Sun, Y., Wang, W.: MEDTO: medical data to ontology matching using hybrid graph neural networks. In: ACM SIGKDD, pp. 2946–2954 (2021)
22. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. arXiv preprint [arXiv:1901.08746](https://arxiv.org/abs/1901.08746) (2019)
23. Gu, Y., Qu, X., Wang, Z., Zheng, Y., Huai, B., Yuan, N.J.: Delving deep into regularity: a simple but effective method for Chinese named entity recognition. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (2022)
24. Donnelly, K.: SNOMED-CT: the advanced terminology and coding system for eHealth. *Stud. Health Technol. Inf.* **121**, 279 (2006)
25. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W., Wright, L.W.: NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Inf.* **40**(1), 30–43 (2007)
26. Hayamizu, T.F., Mangan, M., Corradi, J.P., Kadin, J.A., Ringwald, M.: The adult mouse anatomical dictionary: a tool for annotating and integrating data. *Genome Biol.* **6**(3), 1–8 (2005)
27. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)

28. Jiménez-Ruiz, E., Grau, B.C., Klyne, G., Noy, N.F.: LogMapBio: a scalable approach for biomedical ontology alignment. In: Proceedings of the International Conference on Biomedical Ontologies (ICBO 2021), pp. 102-113. CEUR-WS.org (2021)
29. Noy, N. F., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res.* **37**(suppl_2), W170–W173 (2009)
30. Faria, D., Contreiras Silva, M., Cotovio, P., Eugénio, P., Pesquita, C.: Matcha and matcha-DL results for OAEI 2022. In: OM@ ISWC, pp. 197–201 (2023)