Article

DAEOM: A Deep Attentional Embedding Approach for Biomedical Ontology Matching

Jifang Wu ¹ , Jianghua Lv ^{1,*}, Haoming Guo ² and Shilong Ma ¹

¹ State Key Laboratory of Software Development Environment, Beihang University, Beijing 100091, China; wujifang@buaa.edu.cn (J.W.); slma@nlsde.buaa.edu.cn (S.M.)

² Institute of Software Chinese Academy of Sciences, Beijing 100190, China; haoming@iscas.ac.cn

* Correspondence: jhlv@buaa.edu.cn

Received: 29 September 2020; Accepted: 4 November 2020; Published: 8 November 2020

Abstract: Ontology Matching (OM) is performed to find semantic correspondences between the entity elements of different ontologies to enable semantic integration, reuse, and interoperability. Representation learning techniques have been introduced to the field of OM with the development of deep learning. However, there still exist two limitations. Firstly, these methods only focus on the terminological-based features to learn word vectors for discovering mappings, ignoring the network structure of ontology. Secondly, the final alignment threshold is usually determined manually within these methods. It is difficult for an expert to adjust the threshold value and even more so for a non-expert user. To address these issues, we propose an alternative ontology matching framework called Deep Attentional Embedded Ontology Matching (DAEOM), which models the matching process by embedding techniques with jointly encoding ontology terminological description and network structure. We propose a novel inter-intra negative sampling skill tailored for the structural relations asserted in ontologies, and further improve our iterative final alignment method by introducing an automatic adjustment of the final alignment threshold. The preliminary result on real-world biomedical ontologies indicates that DAEOM is competitive with several OAEI top-ranked systems in terms of F-measure.

Keywords: ontology matching; graph attentional network; semantic similarity; Negative Sampling; final alignment

1. Introduction

Ontology is a highly interoperable, extensible, and scalable mechanism to represent knowledge as a set of concepts within a domain and the relation among those concepts [1]. Life science is one of the most prominent application areas of ontology technology. Many biomedical ontologies such as SNOMED CT [2], the National Cancer Institute Thesaurus (NCI) [3], and the Foundational Model of Anatomy (FMA) [4] have been developed and utilized in real-world systems. Biomedical ontologies provide domain knowledge to support applications such as semantic annotation of biomedical data, knowledge discovery and exchange, data integration, and decision support [5,6]. To integrate and migrate data among applications, it is crucial to first establish mappings between the entities of their respective ontologies. However there are diverse ways to construct an ontology, which leads to varying degrees of heterogeneities [7]. The heterogeneities among different ontologies have limited interoperability across the ontologies. Ontology matching techniques are usually performed to find semantic correspondences between the entity elements of different ontologies to enable interoperability [8]. Two types of correspondences can be distinguished. While approaches generating simple correspondences are limited to matching single entities (i.e., linking a single entity from a source ontology to a single entity of a target ontology), complex matching approaches are able to

generate correspondences which express more complex relationships between entities from different ontologies [9]. In this paper, we focus on matching single entities.

Early studies regarding automatic ontology matching has focused on engineering features from terminological, structural, extensional (individuals of concepts) information, and external resource [10,11]. These features are utilized to compute the similarities of ontological entities (i.e., concepts, properties, individuals) for guiding ontology matching. Up to now, the mainstream ontology matching systems (e.g., AML [12], FCA_Map [13], LogMap [14], and XMap [15]) still use feature-based strategies to evaluate entity similarity. AML employs various sophisticated features and domain-specific thesauruses to perform ontology matching based on heuristic methods that rely on aggregation functions. FCA_Map uses Formal Concept Analysis to derive terminological hierarchical structures that are represented as lattices. The matching is performed by aligning the constructed lattices, taking into account the lexical and structural information that they incorporate. LogMap uses logic-based reasoning over the extracted features and cast ontology matching to a satisfiability problem. XMap relies on the context notion to deal with lexical ambiguity as well as a parallel comparison between concepts to efficiently handle the matching of large ontologies. Feature-based methods mainly employ crafting features of the data to achieve specific tasks. Unfortunately determining which hand-crafted features will be valuable for a given task can be highly time consuming. Cheatham and Hitzler showed that the performance of ontology matching based on such engineered features varies greatly with the domain described by ontologies [16].

As a complement to feature engineering, attempts have been made to develop representation learning techniques for ontology matching. Continuous vectors representing ontological entities can capture the potential associations among features, which is helpful to discover more mappings among ontologies [17].

Representation learning have so far limited impacts on ontology matching, specifically in biomedical ontologies. To the best of our knowledge, only seven literatures have explored the use of representation learning techniques for ontology matching. In [18–20], they used word embeddings [21] to compute the semantic similarities among elements. Zhang et al. [18] firstly introduce word embeddings to the field of ontology matching. They trained word embeddings on Wikipedia and used skip-gram architectures for computing vector representations. They combined edit distance and word embeddings for the computation of the semantic similarities among elements. For textual entity descriptions consisting of more than one word, they proposed a heuristic method to compute semantic similarities among elements. The entity matching strategy was based on maximum similarity. For every entity in the source ontology, the algorithm found the most similar entity in the target ontology. To avoid the problems that come from the semantic similarity and conceptual association coalescence, in [19,20], they used synonymy and antonymy constraints extracted from semantic lexicons to refine pre-trained word embeddings and make them better suited for evaluating semantic similarity. Kolyvakis et al. [19] used the Dual Embedding Space Model (DESM) to compute the semantic similarity of two entity textual description word sets. In [20], they averaged the bag of the word vectors to produce sentence vectors of the entities' textual description, and use the cosine distance to compute the semantic similarity of two entities. In addition, they leverage an outlier detection mechanism based on a denoising autoencoder to improve the performance of alignments. DOME [22] is a scalable matcher that relies on large texts describing ontological concepts. It uses the doc2vec approach to train a fixed-length vector representation of the concepts. Mappings are generated if two concepts are close to each other in the resulting vector space. Xiang et al. [23] proposed an entity representation learning algorithm based on Stacked Auto-Encoders to learn the general representation of the entities. To describe an ontological entity, they design a combination of its ID, labels, and comments. They introduced an iterative similarity propagation method that takes advantage of the more abundant structure information of ontology to discover more mappings. Wang et al. [24] proposed a neural architecture tailored for biomedical ontology matching called OntoEmma, which can encode a variety of information and derive large amounts of labeled data for

training the model. Moreover, they utilize natural language texts associated with entities to further improve the quality of alignments. Li et al. [17] present an alternative ontology matching framework called MultiOM, in which different loss functions were designed based on cross-entropy to model different views among ontologies and learn the vector representations of concepts. They further proposed a novel negative sampling skill tailored for structural relations, which could obtain better vector representations of concepts. After computing the semantic similarity of entities from two compared ontologies with the learned vectors, the final alignment method is executed to select appropriate correspondences between entities of compared ontologies. The final alignment is an important part of the ontology matching process because it directly determines the output result of this process. Several works [17,19,20,23] use the Stable Marriage algorithm [25] for mapping selection during the final alignment. They iteratively pass through all the candidate alignments and discard those with a cosine distance higher than a certain threshold. The adjustment of the threshold value is usually determined manually within matching systems considering the experience of developers.

However, there exist two limitations for the above methods. The first one is the sparsity problem of structural relations. To avoid the poor capability of encoding sparse relations, the above methods prefer terminological-based features to learn concept vectors for discovering mappings, but they do not make full use of structural relations in ontologies. Although Li et al. [17] designed a negative sampling technique to fine-tune the vector representation of concepts by using the network structure indirectly, it can not effectively preserve the network structure. The other is that using the Stable Marriage algorithm during the final alignment has worst-case time complexity $\Theta(n^2 \log n)$ [20]. That is, the larger the ontology, the more time it takes for the final alignment. Meanwhile, the adjustment of the threshold value is usually determined manually within matching systems considering the experience of developers, therefore it is not necessarily adjusted in an optimal way for each pair of compared ontologies [26].

Motivated by the above observations, in this paper we propose a graph attentional autoencoder-based attributed ontology matching framework, which models the matching process by jointly encoding terminological description and network structure to continuous vector representation. Based on the representation, a mapping selection module is proposed to perform the matching algorithm towards better performance. We summarize our contributions as follows:

- We develop the first siamese graph attention-based autoencoder to effectively integrate both network structure and terminological description for deep latent representation learning in ontology matching. To further make full use of structural relations in ontologies, we design a novel inter-intra negative sampling skill tailored for structural relations asserted in ontologies, which can obtain better vector representations of concepts;
- To improve the performance of ontology matching, we combine the Greedy Matching algorithm [27] with the Stable Marriage algorithm to only find the highest correspondences in the semantic similarity matrix as the candidate alignments. We also introduce an automatic adjustment of the threshold method by using the first iteration found for the highest correspondences;
- We implement our method and conduct experiments on real-world biomedical ontologies. The experimental results show that our matching approach can achieve a competitive matching performance compared to several OAEI top-ranked systems in terms of F-measure.

2. Method

2.1. Problem Definition

We consider ontology $O = (C, E, A, X)$, where C is the set of concepts (vertices) and $E \subseteq C \times C$ is the set of relationships (directed edges) between concepts. The topological structure of ontology O can be represented by an adjacency matrix A , where $A_{i,j} = 1$ if $(c_i, c_j) \in E$, otherwise $A_{i,j} = 0$.

$X = \{x_1; \dots; x_n\}$ are the sequence of textual description values where $x_i \in R^d$ is a real-value feature vector associated with concept c_i .

Ontology matching can be formally defined as a function that takes two ontologies O_s and O_t , and returns all semantically equivalent mappings between their different concepts $c_i \in O_s$ and $c_j \in O_t$. In this work, we focus on discovering equivalence correspondences between two ontologies with cardinality 1:1. That is, one concept in ontology O_s can be matched to at most one concept in ontology O_t and vice versa.

2.2. Overall Framework

Existing work [17] designs a negative sampling skill tailored for structural relations to obtain better vector representations of concepts, which only leverages the structural relations for embedding learning indirectly, and can not effectively preserve the network structure information into vectors directly. Graph attention networks (GATs) [28] can assign nodes in a network to low-dimensional representations and effectively preserves the network structure. It is highly effective in network analysis tasks, such as link prediction [29] and node classification [30]. Until recently, this method did not get enough interest in ontology matching. Inspired by their works, we develop an ontology attentional encoder which effectively integrates both structure and terminological description to learn a latent representation. This model uses both direct and indirect methods to solve the sparsity problem of structural relations for representation learning. With jointly encoding terminological description and network structure to continuous vector representation, we can obtain more similar concepts and discover more potential mappings among ontologies. Moreover, there is no need to select and aggregate different single similarities in the similarity computation as described in [17].

Our framework is shown in Figure 1 and consists of two parts: An ontology attentional encoder and a mapping selection module.

- **Ontology attentional encoder.** Our encoder takes the terminological description and network structure as input, and learns the latent embedding by minimizing the reconstruction loss, and designs an inter-intra negative sampling skill to indirectly alleviate the sparsity problem for embedding learning;
- **Mapping Selection.** The mapping selection module performs ontology matching based on the learned representation. It includes two phrases: Candidate selection and final alignment. To reduce the running time of mapping selection, we improve the Stable Marriage algorithm's ability to choose the 1:1 mappings from two different ontologies and introduce an automatic adjustment of the threshold method by using the highest correspondences.

Figure 1. The conceptual framework of Deep Attentional Embedded Ontology Matching (DAEOM).

2.3. Ontology Attentional Encoder

The architecture of the ontology attentional encoder is shown in Figure 2. We first assign each sequence of terminological description associated with concept from source and target ontologies (O_s and O_t) to low-dimensional representations (X_s and X_t) such that semantically similar sequences are close. Then we develop a siamese graph attentional autoencoder, which effectively integrates both network structure (A_s and A_t) and terminological description (X_s and X_t) to learn latent representations Z_s and Z_t . The encoder exploits both network structure and terminological description with a graph attention network, and multiple layers of encoders are stacked to build a deep architecture for

embedding learning. On the other side, the decoder reconstructs the topological network information and manipulates the latent graph representation.

Figure 2. Architecture of ontology attentional encoder.

Terminological Description Embedding. In this paper, we present Terminological-BERT (TBERT), a modification of the Bert [31] network using siamese networks that is able to derive semantically meaningful terminological description embeddings. TBERT adds a pooling operation to the output of BERT to derive a fixed sized terminological description embedding. The pooling strategy is computing the mean of all output vectors (MEAN- strategy). In order to fine-tune BERT, we create siamese networks to update the weights such that the produced terminological description embeddings (X_s and X_t) are semantically meaningful and can be compared with a similarity metric.

We employ a similarity metric, described in [17] (listed as Weizhuo Li), to guide the optimizing procedure. Given input $\{u, v\}$, where u and v are the mapping pair. TBERT seek value of the parameter such that the symmetric similarity metric is large if u and v have the same semantic, and small if they mean different semantics. The similarity metric function $sim_{tb}(u, v)$ that measures the semantic relatedness between mapping pair (u, v) can be defined as:

$$sim_{tb}(u, v) = 2 \cdot \frac{1}{1 + e^{(\|f(u) - f(v)\|_2)}}, \quad (1)$$

where $\|\cdot\|_2$ is the L2-norm, and $f(u)$ and $f(v)$ are the generated terminological description embeddings.

We define a loss function based on cross-entropy to optimize the vector representations of terminological description. The loss function is defined as follows:

$$L_{tb} = - \sum_{(u,v) \in M_{intra}} \log sim_{tb}(u, v) - \sum_{(u,v) \in M'_{intra}} \log (1 - sim_{tb}(u, v)), \quad (2)$$

where M_{intra} is a set of positive mappings and M'_{intra} is a set of negative mappings (see Section 3.2 for details).

Jointly Embedding. After producing all the terminological description embeddings (X_s and X_t) of concepts from source and target ontologies, we present structural-GAT (SGAT), a variant of the graph attention network (GAT) [28] using siamese networks that is able to represent both network structure (A_s and A_t) and terminological description (X_s and X_t) in a unified framework. The idea is to learn hidden representations of each concept that allow for assigning different importance to different concepts within a neighborhood while dealing with different sized neighborhoods to aggregate feature

information. In order to measure the importance of various neighbors, different weights are given to the neighbor representations in our layer-wise graph attention strategy:

$$z_i^{l+1} = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W z_j^l \right). \quad (3)$$

Here, z_i^{l+1} denotes the output representation of concept i , and N_i denotes the neighbors of i . α_{ij} is the attention coefficient that indicates the importance of neighbor concept j to concept i , and σ is a nonlinearity function. To calculate the attention coefficient α_{ij} , we measure the importance of neighbor concept j from both the aspects of the terminological description and topological distance. The attention coefficient α_{ij} can be represented as a single-layer feedforward neural network on the addition of x_i and x_j with weight vector $\vec{a} \in R^{d'}$:

$$c_{ij} = \vec{a}^T [W x_i + W x_j]. \quad (4)$$

Topologically, neighbor concepts contribute towards the representation of a target concept through edges. GAT considers only the 1-hop neighboring concepts (parents and children) for graph attention. In this paper, we assume that two concepts from different ontologies with same parents, children, and siblings considered more similar than concepts that do not have these things in common. Four relationships were used to construct a neighborhood: “self”, “parent”, “child”, and “sibling”. For a given concept, its neighbors include itself, all direct parent and children, and sibling concepts. We obtain a proximity matrix by considering 1-order neighbor concepts (parents and children) and 2-order neighbor concepts (only siblings) in the ontology:

$$P = (B + B^2) / 2. \quad (5)$$

Here B is the transition matrix where $B_{ij} = 1/d_i$ if $r_{ij} \in R$ and $B_{ij} = 0$ otherwise. d_i is the degree of concept i . Therefore P_{ij} denotes the topological relevance of concept j to concept i up to first or second orders. In this case, N_i means the neighboring concepts of i in P . i.e., j is a neighbor of i if $P_{ij} > 0$.

The attention coefficients are usually normalized across all neighborhoods $j \in N_i$ with a softmax function to make them easily comparable across concepts:

$$\alpha_{ij} = \text{softmax}_j(c_{ij}) = \frac{\exp(c_{ij})}{\sum_{r \in N_i} \exp(c_{rj})}. \quad (6)$$

Adding the topological weights P and an activation function σ (here LeakyReLU is used), the coefficients can be expressed as:

$$\alpha_{ij} = \frac{\exp(\sigma P_{ij} (\vec{a}^T [W x_i + W x_j]))}{\sum_{r \in N_i} \exp(\sigma P_{ir} (\vec{a}^T [W x_i + W x_r]))}. \quad (7)$$

We have $x_i = z_i^0$ as the input for our problem, and stack two graph attention layers:

$$z_i^{(1)} = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W^{(0)} x_j \right). \quad (8)$$

$$z_i^{(2)} = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W^{(1)} z_j^{(1)} \right). \quad (9)$$

In this way, our encoder encodes both the structure and terminological description of the concept into a hidden representation, i.e., we will have $z_i = z_i^{(2)}$.

We jointly optimize the ontology autoencoder embedding and matching learning, and define our total objective function as:

$$L = L_r + \gamma L_{sg}, \quad (10)$$

where L_r and L_{sg} are the reconstruction loss of source and target ontologies and matching loss respectively, and $\gamma \geq 0$ is a coefficient that controls the balance in between.

As our latent embedding already contains both content and structure information, we choose to adopt a simple inner product decoder to predict the relationships between concepts, which would be efficient and flexible:

$$\hat{A}_{ij}^s = \text{sigmoid} \left(z_i^{sT} z_j^s \right), \quad (11)$$

$$\hat{A}_{ij}^t = \text{sigmoid} \left(z_i^{tT} z_j^t \right), \quad (12)$$

where $\hat{A}^s$ is the reconstructed structure matrix of the source ontology O_s , and $\hat{A}^t$ is the reconstructed structure matrix of the target ontology O_t .

We minimize the reconstruction error by measuring the difference between A and $\hat{A}$:

$$L_r = \|A_s - \hat{A}_s\|_2 + \|A_t - \hat{A}_t\|_2. \quad (13)$$

Ontology mapping pairs are fed to train the siamese graph attention networks. Given an input $\{cs_i^k, ct_j^k\}$, where cs_i^k and ct_j^k are the k^{th} mapping pair, the i^{th} concept from O_s , and j^{th} concept from O_t . The similarity metric function $sim_{sg}(cs_i^k, ct_j^k)$ that measures the semantic relatedness between mapping pair (cs_i^k, ct_j^k) can be defined as:

$$sim_{sg}(cs_i^k, ct_j^k) = 2 \cdot \frac{1}{1 + e^{(\|Z_i^s - Z_j^t\|_2)}}, \quad (14)$$

where $\|\cdot\|_2$ is the L2-norm.

We define a loss function based on cross-entropy to optimize the vector representations of terminological description. The loss function is defined as follows:

$$L_{sg} = - \sum_{(cs_i^k, ct_j^k) \in M_{inter}} \log sim_{sg}(cs_i^k, ct_j^k) - \sum_{(cs_i^k, ct_j^k) \in M'_{inter}} \log (1 - sim_{sg}(cs_i^k, ct_j^k)), \quad (15)$$

where M_{inter} is a set of positive mappings and M'_{inter} is a set of negative mappings (see Section 3.2 for details).

The model is trained by minimizing the overall loss function. In this paper, we only consider small ontology structure data, so the batch size during model training is 1. Large-scale ontology will be discussed in future research.

2.4. Inter-Intra Negative Sampling

In this paper, the unique `rdfs:label` that accompanies every type in the source and target ontologies is used as a terminological description of the concept. If a concept contains multiple labels, we will choose the first label as the terminological description of the concept. To obtain more candidate mappings for representation learning, we assume that the mappings generated by equivalent strings or their synonym labels are positive samples. It means that the positive mappings M has two parts: A set of positive mappings from inside the ontology (M_{intra}) and a set of positive mappings from inter-ontology (M_{inter}). Each concept contains native attributes: Name, alias, and synonyms, these attributes have the same semantics. We construct M_{intra} and M_{inter} through these attributes. An example can be seen in Figure 3.

Furthermore, we design a novel inter-intra negative sampling tailored for structural relations (e.g., subclassOf relations) asserted in ontologies, which can obtain better vector representations of entities for ontology matching. This method can indirectly solve the sparsity problem of structural relations like [17]. Unlike the uniform negative sampling method that samples its replacer from all the concepts, we limit the sampling scope to a group of candidates. The negative mappings $M' = M'_{intra} \cup M'_{inter}$. For a given positive sample $(c_i, c_j) \in M_{intra}$, if there exist subclassOf relations (e.g., $(c'_i, \text{subclassOf}, c_j)$ or $(c'_j, \text{subclassOf}, c_i)$) asserted in ontologies, we need to exclude this replace case. If concept u of ontology O_s has a similar semantic with entity v of ontology O_t , and u can not match with another concept of ontology O_t . In addition, v can not match with another concept of ontology O_s . For a given positive sample $(u, v) \in M_{inter}$, we corrupt it and randomly replace u or v to generate negative mapping pairs. Unlike the uniform negative sampling method that samples its replacer from all the concepts, we randomly select a concept that was not the neighbor of u and v . That is, the randomly selected concept is not a parent or child of u and v , because the subclass of relationship of our datasets are transitive [32].

Figure 3. An example of constructing inter-intra candidate positive mappings.

2.5. Mapping Selection

As shown in Figure 4, the mapping selection consists of two phrases. One is a candidate selection (in blue boxes), another is the final alignment (in green boxes). We compute the distance using the euclidean distance over concept embeddings learned by jointly embedding the model. We iteratively match the concepts of two different ontologies using the improved Stable Marriage algorithm (SM) over the concept embeddings' pairwise distances. We also introduce an automatic adjustment of the threshold method by using the first iteration found in the highest correspondences. Finally, we pass through all the candidate alignments and discard those with a euclidean distance higher than a threshold.

As the computation of the preference matrix required for defining the Stable Marriage assignment problem's instance has worst-case time complexity $\Theta(n^2 \log n)$ [20], it means that aligning larger ontologies will consume much more time. To improve the performance of ontology matching, we combine the Greedy Matching algorithm with the Stable Marriage algorithm to reduce the runtime of ontology matching. Inspired by [26], we only find the highest correspondences in the semantic similarity matrix as the candidate alignments. A correspondence between concept c_s of ontology O_s and c_t of ontology O_t is the highest correspondence if and only if it has the smallest semantic distance than any other correspondence of either c_s or c_t with some other concept. We iteratively find the highest correspondences from the semantic similarity matrix until there is no highest correspondence or concepts are all matched in either O_s or O_t . In this way, we do not have to find the most similar concepts for all concepts. Experiments show that only finding the highest correspondences can reduce the running time without affecting the matching performance.

We iteratively pass through all the candidate alignments and discard those with a euclidean distance higher than a certain threshold t . The adjustment of the threshold value is usually

determined manually within matching systems considering the experience of developers. In this paper, we introduce an automatic adjustment of the threshold method instead of manual adjustment. Since the highest correspondences found in the first iteration are most likely to be the correct matching pairs, we use their semantic similarity to calculate the threshold t [26]. The threshold value is defined as:

$$t = \bar{Q} + SD_Q, \quad (16)$$

where $Q = \{q_1, q_2, \dots, q_i\}$ ($i > 0$) is the semantic similarity of all highest correspondences found in the first iteration. $\bar{Q}$ is the arithmetic mean of Q and SD_Q is the standard deviation of Q .

The algorithm of the whole module is shown in Algorithm 1.

Algorithm 1: Mapping selection

Input: The concept list of source and target ontology: C_s and C_t , The hidden representation of source and target ontology: Z_s and Z_t ;

Output: Matching results R ;

```

1 compute similarity matrix:  $S = \text{euclidean distance}(Z_s, Z_t)$ ;
2 find the smallest semantic distance for each concept in source ontology:
    $M_s = \text{argmin}(S, \text{axis} = 0)$ ;
3 find the smallest semantic distance for each concept in target ontology:
    $M_t = \text{argmin}(S, \text{axis} = 1)$ ;
4  $\text{count} = 0$ ;
5  $R = []$ ;
6  $t = 0$ ;
7 while  $\text{len}(R) \neq \min(\text{len}(C_s), \text{len}(C_t))$  do
8    $HC = \text{findHighestCorrespondence}(M_s, M_t, C_s, C_t)$ ;
9    $\text{count}++$ ;
10  if  $\text{count} == 1$  then
11     $Q = HC$ ;
12     $t = \bar{Q} + SD_Q$ ;
13  end
14  foreach  $\text{iter} \in HC$  do
15    if the semantic similarity in  $\text{iter} \leq t$  then
16      put the founded matching into  $R$ ;
17      modify the row and column similarity values of matched concepts in  $S$  to be  $+\infty$ ;
18    else
19      modify the similarity value of those two concepts in  $S$  to be  $+\infty$ ;
20    end
21  end
22  find the smallest semantic distance for each concept in source ontology:
    $M_s = \text{argmin}(S, \text{axis} = 0)$ ;
23  find the smallest semantic distance for each concept in target ontology:
    $M_t = \text{argmin}(S, \text{axis} = 1)$ ;
24 end
25 return  $R$ ;
```

Figure 4. Overall proposed ontology matching architecture.

3. Experiments and Discussion

To verify the effectiveness of DAEOM, we used Python to implement our approach with the aid of TensorFlow. All reported experiments were performed on a desktop computer with an Intel® Core™ i7-6700K (3.40 GHz) processor with 32 GB RAM and one NVIDIA® GeForce® GTX™ 1080 (8 GB) graphic card.

3.1. Dataset

The experiments were performed on biomedical evaluation benchmarks coming from the Ontology Alignment Evaluation Initiative (OAEI), which organizes annual campaigns for evaluating ontology matching systems.

Biomedical ontologies were collected from the Anatomy Track and Large BioMed Track in OAEI. We used four ontologies in our ontology mapping experiments, which are Adult Mouse Anatomical Dictionary (MA) [33], NCI Thesaurus (NCI), Foundational Model of Anatomy (FMA), and SNOMED Clinical Terms (SNOMED). Two of them (FMA and MA) are pure anatomical ontologies, while the other two (SNOMED and NCI) are broader biomedical ontologies, and anatomical structure is their subdomain [34]. These resources are available in the Ontology Alignment Evaluation Initiative. We provide some details regarding the respective size of each ontology matching task in Table 1. The ontology matching tasks are MA-NCI, FMA-NCI, and FMA-SNOMED. The “Nodes” is the ontology entities. The “Relations” is the edges (SUBCLASS-OF) between nodes in the ontology. In this paper, we only consider the SUBCLASS-OF relationship of these datasets. We have only focused on discovering one-to-one matchings. The “Matchings” only include the nodes one-to-one types’ equivalences.

Table 1. Respective sizes of the ontology matching tasks.

Task	O _s :Nodes	O _s :Relations	O _t :Nodes	O _t :Relations
MA ¹ -NCI ²	2737	1807	3298	3761
FMA ³ -NCI	3696	3693	6488	4917
FMA-SNOMED ⁴	10,157	10,154	13,412	16,287

¹ Adult Mouse Anatomical Dictionary. ² NCI Thesaurus. ³ Foundational Model of Anatomy. ⁴ SNOMED Clinical Terms.

To represent each ontology node by jointly encoding its textual description and network structure information, the unique rdfs:label that accompanies every type in the ontologies is used as a textual description of the node. We performed a textual preprocessing, including case-folding, tokenization, removal of English stopwords, and words coappearing in terms (for example, the word “structure” in SNOMED). The reference alignments of these alignment scenarios are based on the UMLS Metathesaurus [35], which currently consists of the most comprehensive effort for integrating

independently developed medical thesauri and ontologies. All the mappings marked as “?” in the reference alignment were considered as positive.

3.2. Evaluation Measures

We followed the standard evaluation criteria from the OAEL, calculating the precision (P), recall (R), and F-measure (F) over each test. Given a reference alignment G , the precision and recall of some alignment A are defined as (17) and (18):

$$P = \frac{|G \cap A|}{|A|} \quad (17)$$

$$R = \frac{|G \cap A|}{|G|}. \quad (18)$$

The F-measure of some alignment A is basically the harmonic means of precision and recall, defined as (19):

$$F = \frac{2 \times P \times R}{P + R}. \quad (19)$$

3.3. Experiment Settings

We select several strategies to construct the baseline methods to explore the performance details of our algorithm. The following is the detailed construction of strategies in our experiments.

- OM-TD (TF-IDF [36]): It uses TF-IDF to calculate the similarity of ontology terminological description by the procedure described in [17];
- OM-TD (LSTM [37]): It only uses LSTM to produce ontology terminological description embeddings ignoring ontology network structure information for ontology matching;
- OM-TD (TBERT): It employs our proposed ontology terminological description embedding component without encoding network structure information;
- OM (LSTM + SGAT): It extends the LSTM experiment by applying our proposed SGAT to jointly encoding terminological description and network structure for ontology matching;
- OM (TBERT + GraphSAGE [38]): It combines our proposed TBERT and GraphSAGE to jointly encode a terminological description and network structure for ontology matching;
- OM (TBERT + TransE [39]): It employs TransE, jointly encoding terminological description and network structure for ontology matching. The vectors of nodes are initialized by our proposed TBERT;
- OM (DAEOM): Is our proposed jointly embedding method for ontology matching.

“OM-TD” means only using terminological description for ontology matching. “OM” means using both network structure and terminological description. Those methods applied our proposed mapping selection module in final alignment.

For DAEOM, we use Adam as an optimizer and the configuration of hyper-parameters is listed below: Word vector has the size (d) 200 and is shared everywhere. We initialized the word vectors from word vectors pre-trained on a combination of PubMed and PMC texts with texts extracted from a recent English Wikipedia dump [40]. All the initial out-of-vocabulary word vectors are sampled from a normal distribution ($\mu = 0, \sigma^2 = 0.01$). We selected $\{2, 4, 6\}$ negative triples sampled for each positive triple. In TBERT, the number of block N and multi-head h_e in transformer-encoder were 2 and 4. The whole training of TBERT spent 50 epochs. The mini-batch size was set to $Tbatch = \{100, 500, 1000\}$. We selected the learning rate λ_{tb} among $\{0.01, 0.001, 0.0001\}$. In SGAT, we set the matching coefficient γ to 10. The output concept vector had size (d') 25. The SGAT was trained over 100 epochs. The mini-batch size was set to $Sbatch = 1$. We selected the learning rate λ_{sg} among $\{0.01, 0.001, 0.0001\}$.

In order to show the effect of our proposed negative sampling, a symbol “-” added to the symbol represented a module that indicates that this module is not equipped with negative sampling tailored for structural relations.

3.4. Results and Discussion

Table 2 lists the matching results of DAEOM compared with baseline systems. We can see that our method clearly outperforms all the baselines across all the evaluation metrics. Comparing the F-measure of TF-IDF and our proposed TBERT, the latter achieves a higher performance than TF-IDF. The main reason for this is that continuous vectors representing tokens can provide more semantic information than single strings for calculating the similarity of concepts. To validate the importance of our ontology term encoder component (TBERT), we further analyzed the behavior of aligning ontologies based on the LSTM. As we can see, TBERT achieves a statistically significant higher performance than LSTM in all experiments. This behavior indicates the superiority of TBERT in injecting semantic similarity to terminological description embeddings. Moreover, the performance of DAEOM is better than TBERT in term of F-measure. We further extended the LSTM experiment by applying our proposed SGAT. This extension leads to the same effects as the ones summarized in the TBERT–DAEOM comparison. This observation demonstrates that both the network structure and terminological description contain useful information for ontology matching, and illustrates the significance of capturing the interplay between two-sides information. As our proposed model is graph-based, we also select two graph-based models GraphSAGE and TransE as the baselines. Notice that the performance of DAEOM is much more better than combining TBERT with GraphSAGE or TransE. The main reason is that the GAT can learn node embeddings that allow for assigning different importance to different nodes within a neighborhood while dealing with different sized neighborhoods. Moreover, the performance of DAEOM is better than DAEOM in terms of the F-measure. This further proves that employing structural relations are helpful to distinguish the vector representations of concepts.

Table 2. The comparison of DAEOM with baseline methods.

Method	MA-NCI			FMA-NCI (Small)			FMA-SNOMED (Small)		
	P	R	F	P	R	F	P	R	F
OM-TD (TF-IDF)	0.900	0.648	0.785	0.969	0.734	0.835	0.941	0.613	0.742
OM-TD (LSTM)	0.968	0.704	0.815	0.958	0.871	0.912	0.972	0.687	0.805
OM-TD (TBERT)	0.977	0.702	0.817	0.966	0.878	0.920	0.977	0.715	0.826
OM (LSTM + SGAT)	0.975	0.717	0.826	0.971	0.879	0.923	0.981	0.732	0.838
OM (TBERT + GraphSAGE)	0.954	0.529	0.681	0.981	0.738	0.843	0.913	0.677	0.777
OM (TBERT + TransE)	0.890	0.502	0.642	0.961	0.558	0.706	0.722	0.506	0.595
OM (DAEOM-)	0.977	0.736	0.840	0.984	0.871	0.924	0.987	0.783	0.873
OM (DAEOM)	0.981	0.748	0.849	0.989	0.888	0.936	0.990	0.791	0.879

Note: “OM-TD” means only using terminological description for ontology matching. “OM” means using both network structure and terminological description. Bold numbers indicate the best performance on each matching task, respectively.

Table 3 lists a comparison of DAEOM with six top-performing systems based on feature engineering and representation learning, according to the results published in the Anatomy track and Large BioMed track by OAEI 2019. The preliminary result shows that DAEOM obtains the top performance in the two ontology mappings tasks (FMA-NCI (small)) and FMA-SNOMED (small)). In those two, the precision and F-measure are significantly better than all other systems. In terms of recall, DAEOM demonstrates lower performance in the ontology matching tasks. However, we would like to note that we have not used any semantic lexicons specific to the biomedical domains compared to the other systems. For instance, AML uses three sources of biomedical background knowledge to extract synonyms. Specifically, it exploits the Uber Anatomy Ontology (Uberon), the Human Disease

Ontology (DOID), and the Medical Subject Headings (MeSH) [12]. Hence, our reported recall can be explained due to the lower coverage of biomedical terminology in the semantic lexicons that we have used. Nevertheless, there still exists a gap compared with the best systems (e.g., AML, LogMap, POMAP++) in MA-NCI. We analyze that only considering the SUBCLASS-OF relationship may be the main reason. In MA-NCI, the ontology O_s contains a large number of property relationships, which we did not use. This is also the reason why the relations of ontology O_s of this task in Table 1 is sparser than that of other ontology. We leave this issue in our future work.

In Table 4 we demonstrate a sample of results produced by aligning ontologies using improved Stable Marriage's solution based on a similarity matrix computed either on DAEOM or OM-TD(LSTM) in FMA-NCI task. The automatically calculated threshold of DAEOM is 0.046, and the other is 0.097. These results are divided into a true and false alignment. As we know, there are two types of ontological heterogeneity. The first one is that different concepts from the two ontologies contain the same semantics, but are defined as different name labels. The other is that two concepts are defined as the same name labels but their semantemes are very different. From those results in Table 4, it can be seen that DAEOM can effectively distinguish two concepts that have name or semantic heterogeneity by jointly encoding ontology terminological description and network structure. In particular, our method is more effective than the term-only method to match concepts that have numbers and abbreviations in their name labels.

Table 3. The comparison of DAEOM with OAEI 2019 top-ranked systems.

Method	MA-NCI			FMA-NCI (Small)			FMA-SNOMED (Small)		
	P	R	F	P	R	F	P	R	F
AML	0.950	0.936	0.943	0.958	0.910	0.933	0.923	0.762	0.835
LogMap	0.918	0.846	0.880	0.944	0.897	0.920	0.947	0.690	0.798
LogMapBio	0.872	0.925	0.898	0.919	0.912	0.915	0.931	0.703	0.801
POMAP++	0.919	0.877	0.897	0.979	0.814	0.889	0.906	0.260	0.404
FCAMapKG	0.996	0.631	0.772	0.967	0.817	0.886	0.973	0.222	0.362
DOME	0.993	0.615	0.760	0.985	0.764	0.861	0.988	0.198	0.330
DAEOM	0.981	0.748	0.849	0.989	0.888	0.936	0.990	0.791	0.879

Note: Bold numbers indicate the best performance on each matching task, respectively.

Table 4. Sample results produced by aligning ontologies using either DAEOM or OM-TD(LSTM) in FMA-NCI task. The threshold of DAEOM is 0.046, and the other is 0.097.

Ground truth	Concept from O_s	Concept from O_t	LSTM Similarity	DAEOM Similarity
True	internal carotid artery	internal carotid artery	0.000	0.009
	lamina propria bronchus	bronchus lamina propria	0.094	0.012
	vitreous chamber eyeball	vitreous chamber	0.162	0.035
	ligamentum teres liver	round ligament liver	0.101	0.009
	cerebellomedullary cistern	cistern magna	0.291	0.013
	photoreceptor layer retina	layer rod cone	0.242	0.012
	root tooth	radix dentis	0.192	0.011
	eotaxin 2	small inducible cytokine a24	0.601	0.013
	fourth lumbar vertebra	l4 vertebra	0.247	0.012
False	dentition	dentition	0.000	0.064
	adherens junction	zonula adherens	0.084	0.113
	masticatory muscle	masseter muscle	0.078	0.114

3.5. Runtime Analysis

In this section, we report the runtimes of our ontology matching algorithm for different matching scenarios. Since DAEOM consists of three major steps, we present the time devoted to each of them

in Figure 5a. In brief, the steps of our algorithm are the following: The training of TBERT (Step 1), the training of SGAT (Step 2), and the mapping selection (Step 3).

Figure 5. (a) Runtimes of the steps in the proposed algorithm. (b,c) Analysis of the running time and F-measure of improved Stable Marriage (SM) matching.

As seen in Figure 5a, the majority of the time is allotted to the training of TBERT and SGAT. Specifically, we would like to note that training time depends on the number of epoch. Besides, it can be observed that running time gradually increases as the scale of ontology. We also experimented to analyze the running time and performance (F-measure) of our improved Stable Marriage (SM) matching across the different matching scenarios. The experimental results are shown in Figure 5b,c. With the increasing scale of ontology, the running time of the unimproved matching presents exponential growth. However, the running time of our improved matching is linear growth. In addition, we can see that the running time of the improved matching is shorter than unimproved matching across different matching scenarios. This can be explained as follows: The unimproved solution requires each entity of ontology O_s to be compared across all possible entities of ontology O_t , and has worst-case time complexity $\Theta(n^2 \log n)$. Our improved matching only finds the highest correspondences in the semantic similarity matrix as candidate alignments. The time complexity is $\Theta(n)$. In Figure 5c, we can see that the improved and unimproved matching methods obtain similar or even better performances in all different matching tasks. Experimental results show that only finding the highest correspondences can reduce the running time without affecting the matching performance.

3.6. Threshold

The threshold t constitutes a means for quantifying if two entities are semantically similar. To validate our automatic adjustment threshold method is effective, we perform a sensitivity analysis for the threshold t and compare between the automated threshold and manual threshold about performance (F-measure).

Figure 6a shows a threshold sensitivity analysis of our method across the different ontology matching tasks. For exploring the effectiveness of threshold t , we present the performance of DAEOM for all different matching scenarios when manually varying the value of t between 0 and 1 at 0.1 intervals. It is observed that the performance (F-measure) monotonically increases when the value of t varies between 0 and approximately 0.1. Then the performance decreases with $t \in [0.1, 0.6]$ and reaches an asymptotic value at about 0.6.

We report the performance (F-measure) results with automated and manual threshold across the different matching tasks in Figure 6b. As the performance increases with the value of t varying between 0 and approximately 0.1, we only manually change the threshold between 0 and 0.1 at 0.01 intervals. It can be observed that the automated threshold values are closed to the optimal values of the manual threshold. They obtain similar or even better performances in all the different matching

tasks. The experimental results prove that using the highest correspondences found in the first iteration to calculate the threshold is effective for ontology matching.

Figure 6. (a) Sensitivity analysis of the performance (F-measure) with different threshold values. (b) Comparative analysis of the automated threshold and manual threshold about performance (F-measure).

4. Conclusions

In this paper, we developed siamese graph attention-based autoencoders to effectively integrate both network structure and terminological description for deep latent representation learning in ontology matching. We based our ontology embedding architecture on Bert, Graph Attention Networks, and Siamese Neural Networks. We computed ontologies semantic similarity using the euclidean distance over the node embedding vectors. We found the highest correspondences in the semantic similarity matrix as the candidate alignments to reduce the runtime of ontology matching. Finally, we use the highest correspondences found in the first iteration to automate the threshold instead of manual adjustment. Experimental results on the datasets from OAEI demonstrated that our approach performed better than most of the participants and achieved competitive performance.

Nonetheless, our approach also has certain shortcomings. To begin with, our approach employs one single model for all relations without distinction, which inevitably restricts the capability of network embedding. We only considered the SUBCLASS-OF relationship in the ontology. In fact, there are many types of relationships in ontology. A promising direction for future research is to investigate heterogeneous relations of ontology. At the same time, we also only considered small ontology matching. It means that graph segmentation and distributed model training did not need to be considered. Large-scale ontology will be discussed in future research.

Author Contributions: Conceptualization, J.W.; Data curation, J.W.; Formal analysis, H.G.; Methodology, J.W.; Software, J.W.; Validation, J.W., J.L. and S.M.; Writing—original draft, J.W.; Writing—review and editing, J.W., J.L. and H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ramis, B.; Gonzalez, L.; Iarovyi, S.; Lobov, A.; Lastra, J.L.M.; Vyatkin, V.; Dai, W. Knowledge-based web service integration for industrial automation. In Proceedings of the 2014 12th IEEE International Conference on Industrial Informatics (INDIN), Porto Alegre, Brazil, 27–30 July 2014; pp. 733–739.
2. Donnelly, K. SNOMED-CT: The advanced terminology and coding system for eHealth. *Stud. Health Technol. Inform.* **2006**, *121*, 279–290. [[PubMed](#)]
3. De Coronado, S.; Haber, M.W.; Sioutos, N.; Tuttle, M.S.; Wright, L.W. NCI Thesaurus: Using science-based terminology to integrate cancer research results. *Stud. Health Technol. Inform.* **2004**, *107*, 33–37. [[PubMed](#)]
4. Rosse, C.; Mejino, J.L., Jr. A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *J. Biomed. Inform.* **2003**, *36*, 478–500. [[CrossRef](#)] [[PubMed](#)]

5. Bodenreider, O. Biomedical ontologies in action: Role in knowledge management, data integration and decision support. *Yearb. Med. Inform.* **2008**, *17*, 67–79. [\[CrossRef\]](#)
6. Hoehndorf, R.; Schofield, P.N.; Gkoutos, G.V. The role of ontologies in biological and biomedical research: A functional perspective. *Brief. Bioinform.* **2015**, *16*, 1069–1080. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Xie, C.; Chekol, M.W.; Spahiu, B.; Cai, H. Leveraging Structural Information in Ontology Matching. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 1108–1115.
8. Shvaiko, P.; Euzenat, J. Ontology Matching: State of the Art and Future Challenges. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 158–176. [\[CrossRef\]](#)
9. Thiéblin, E.; Haemmerlé, O.; Hernandez, N.; Trojahn, C. Survey on complex ontology matching. *Semant. Web* **2019**, 1–39, Preprint.
10. Cheatham, M.; Pesquita, C. Semantic data integration. In *Handbook of Big Data Technologies*; Zomaya, A.Y., Sakr, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 263–305.
11. Nezhadi, A.H.; Shadgar, B.; Osareh, A. ontology alignment using machine learning techniques. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *3*, 139–150.
12. Faria, D.; Pesquita, C.; Santos, E.; Palmonari, M.; Cruz, I.F.; Couto, F.M. The AgreementMakerLight Ontology Matching System. In *OTM Confederated International Conferences on the Move to Meaningful Internet Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 527–541.
13. Zhao, M.; Zhang, S.; Li, W.; Chen, G. Matching biomedical ontologies based on formal concept analysis. *J. Biomed. Semant.* **2018**, *9*, 11. [\[CrossRef\]](#)
14. Jiménez-Ruiz E.; Grau, B.C. LogMap: Logic-based and Scalable Ontology Matching. In *International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 273–288.
15. Djeddi, W.E.; Khadir, M.T. A novel approach using context-based measure for matching large scale ontologies. In *International Conference on Data Warehousing and Knowledge Discovery*; Springer: Cham, Switzerland, 2014; pp. 320–331.
16. Cheatham, M.; Hitzler, P. String similarity metrics for ontology alignment. In *International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 294–309.
17. Li, W.; Duan, X.; Wang, M.; Zhang, X.; Qi, G. Multi-view Embedding for Biomedical Ontology Matching. *OM@ ISWC 2019*, 2536, 13–24.
18. Zhang, Y.; Wang, X.; Lai, S.; He, S.; Liu, K.; Zhao, J.; Lv, X. Ontology Matching with Word Embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*; Springer: Cham, Switzerland, 2014; pp. 34–45.
19. Kolyvakis, P.; Kalousis, A.; Kiritsis, D. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, 1–6 June 2018; pp. 787–798.
20. Kolyvakis, P.; Kalousis, A.; Smith, B.; Kiritsis, D. Biomedical ontology alignment: an approach based on representation learning. *J. Biomed. Semant.* **2018**, *9*, 21–40. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Curran: Lake Tahoe, NV, USA, 2013; pp. 3111–3119.
22. Hertling, S.; Paulheim, H. “DOME results for OAEI 2019.” *OM@ ISWC 2019*, 2536, 123–130.
23. Xiang, C.; Jiang, T.; Chang, B.; Sui, Z. ERSOM: A Structural Ontology Matching Approach Using Automatically Learned Entity Representation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2419–2429.
24. Wang, L.L.; Bhagavatula, C.; Neumann, M.; Lo, K.; Wilhelm, C.; Ammar, W. Ontology alignment in the biomedical domain using entity definitions and context. *arXiv* **2018**, arXiv:1806.07976.
25. Gale, D.; Shapley, L.S. College Admissions and the Stability of Marriage. *Am. Math. Mon.* **1962**, *69*, 9–15. [\[CrossRef\]](#)
26. Gulić, M.; Vuković, M. An Iterative Automatic Final Alignment Method in the Ontology Matching System. *J. Inf. Organ. Sci.* **2018**, *42*, 39–61. [\[CrossRef\]](#)
27. Dyer, M.; Frieze, A.; Pittel, B. The average performance of the greedy matching algorithm. *Ann. Appl. Probab.* **1993**, *3*, 526–552. [\[CrossRef\]](#)

28. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903.
29. Lü, L.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A Stat. Mech. Appl.* **2011**, *390*, 1150–1170. [[CrossRef](#)]
30. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *AI Mag.* **2008**, *29*, 93. [[CrossRef](#)]
31. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Kotitsas, S.; Pappas, D.; Androutsopoulos, I.; McDonald, R.; Apidianaki, M. Embedding Biomedical Ontologies by Jointly Encoding Network Structure and Textual Node Descriptors. *arXiv* **2019**, arXiv:1906.05939.
33. Hayamizu, T.F.; Mangan, M.; Corradi, J.P. The Adult Mouse Anatomical Dictionary: A tool for annotating and integrating data. *Genome Biol.* **2005**, *6*, 29–36. [[CrossRef](#)]
34. Zhang, S.; Bodenreider, O. Experience in Aligning Anatomical Ontologies. *Int. J. Semant. Web Inf. Syst.* **2007**, *3*, 1–26. [[CrossRef](#)]
35. Olivier, B. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Res.* **2004**, *32*, 267–270.
36. Aizawa, A. An information-theoretic perspective of tf-idf measures. *Inform. Process. Manag.* **2003**, *39*, 45–65. [[CrossRef](#)]
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
38. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*; Curran: Long Beach, CA, USA, 2017; pp. 1024–1034.
39. Bordes, A.; Usunier, N.; Garcia-Duran, A. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*; Curran: Lake Tahoe, NV, USA, 2013; pp. 2787–2795.
40. Pysalo, S.; Ginter, F.; Moen, H.; Salakoski, T.; Ananiadou, S. Distributional semantics resources for biomedical text processing. In *Proceedings of the The 5th International Symposium on Languages in Biology and Medicine (LBM 2013)*, Tokyo, Japan, 12–13 December 2013; Database Center for Life Science: Tokyo, Japan, 2013; pp. 39–44.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).