Multi-view Embedding for Biomedical Ontology Matching

Weizhuo Li[1,2], Xuxiang Duan[3], Meng Wang[1,2], XiaoPing Zhang[4(✉)], and Guilin Qi[1,2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China.
liweizhuo@amss.ac.cn, {meng.wang,gqi}@seu.edu.cn
[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China.
[3] School of Mathematical Sciences, Chongqing Normal University, Chongqing, China.
duanxx9156@163.com
[4] China Academy of Chinese Medical Sciences, Beijing, China.
xiao_ping_zhang@139.com

Abstract. The goal of ontology matching (OM) is to identify mappings between entities from different yet overlapping ontologies so as to facilitate semantic integration, reuse and interoperability. Representation learning methods have been applied to OM tasks with the development of deep learning. However, there still exist two limitations. Firstly, these methods are of poor capability of encoding sparse entities in ontologies. Secondly, most methods focus on the terminological-based features to learn word vectors for discovering mappings, but they do not make full use of structural relations in ontologies. It may cause that these methods heavily rely on the performance of pre-training and are limited without dictionaries or sufficient textual corpora. To address these issues, we propose an alternative ontology matching framework called MultiOM, which models the matching process by embedding techniques from multiple views. We design different loss functions based on cross-entropy to learn the vector representations of concepts, and further propose a novel negative sampling skill tailored for the structural relations asserted in ontologies. The preliminary result on real-world biomedical ontologies indicates that MultiOM is competitive with several OAEI top-ranked systems in terms of F1-measure.

Key words: Ontology Matching, Embedding, Cross-Entropy, Negative Sampling

1 Introduction

In the Semantic Web, ontologies aim to model domain conceptualizations so that applications built upon them can be compatible with each other by sharing the same meanings. Life science is one of the most prominent application areas of ontology technology. Many biomedical ontologies have been developed and utilized in real-world systems including Foundational Model of Anatomy (FMA)[5], Adult Mouse Anatomy (MA) for anatomy[6], National Cancer Institute Thesaurus (NCI)[7] for disease and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT)[8] for clinical medicine. To

[5] http://si.washington.edu/projects/fma

[6] http://informatics.jax.org/vocab/gxd/ma_ontology

[7] https://ncit.nci.nih.gov/

[8] http://www.snomed.org/snomed-ct/

integrate and migrate data among applications, it is crucial to first establish mappings (or correspondences) between the entities of their respective ontologies. As ontologies in the same domain are often developed for various purposes, there exist several differences in coverage, granularity, naming, structure and many other aspects. It severely impedes the sharing and reuse of ontologies. Therefore, ontology matching (OM) techniques devote to identify mappings across ontologies in order to alleviate above heterogeneities [1].

In the last ten years, many automatic systems are developed so as to discover mappings between independently developed ontologies and obtain encouraging results (see [2, 3] for a comprehensive and up-to-date survey). Up to now, the mainstream methods (e.g., LogMap [4], AML [5], FCA-Map [6], XMap [7]) still focus on engineering features from terminological, structural, extensional (individuals of concepts) information and external resource [1]. These features are utilized to compute the similarities of ontological entities (i.e., concepts, properties, individuals) for guiding the ontology matching. With the development of deep learning [8], there also exist several works (e.g., ERSOM [9], DeepAlignment [10], SCBOW + DAE(O) [11] OntoEmma [12]) that try to shift from feature engineering to representation learning. The assumption is that semantically similar or related words appear in similar contexts. Therefore, word vectors own the potentials that can bring significant value to OM given the fact that a great deal of ontological information comes in textual form [10]. Nevertheless, there still exist two challenges that need to be solved:

- **Sparsity Problem for Embedding Learning**: One of the main difficulties for embedding learning is of poor capability of encoding sparse entities. Even in large-scale medical ontologies with lots of relations, most knowledge graph embedding techniques (e.g., TransE [13]) are still not applicable. Zhang et al. [14] observed that the prediction results of entities were highly related to their frequency, and the results of sparse entities were much worse than those of frequent ones.
- **Limitation Problem for External Resource**: Thesaurus is one kind of external resource that is usually employed in matching systems such as WordNet [15], UMLS Metathesaurus[9]. In addition, textual descriptions can also be employed for ontology matching [11, 12]. Nevertheless, these methods based on representation learning rely heavily on the performance of pre-training. Therefore, it may limit their scalability if there exist no dictionaries or sufficient textual corpora.

To address above problems, we propose MultiOM, an alternative ontology matching framework based on embedding techniques from multiple views. The underlying idea is to divide the process of OM into different modules (i.e., lexical-embedding module, structural-embedding module, resource-embedding module) and employ embedding techniques to soften these modules. Existing works [16–18] show that identifying multiple views can sufficiently represent the data and improve the accuracy and robustness of corresponding tasks. Therefore, we employ this idea to characterize the process of OM and try to alleviate the sparsity problem for embedding learning indirectly. More precisely, different loss functions are designed based on cross-entropy to model different views among ontologies and learn the vector representations of ontological entities. With continuous vector representation, we can obtain more similar concepts and discover more potential mappings among ontologies. We further treat ontologies as external resources instead of textual descriptions and thesaurus. Compared with these resources, ontologies own structural assertions naturally that can refine the quality of

[9] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html

discovered mappings [19]. Furthermore, we design a novel negative sampling tailored for structural relations (e.g., *subclassOf* relations, *disjointWith* relations) asserted in ontologies, which can obtain better vector representations of entities for OM.

The contributions of our study are summarized as follows.

- We propose an alternative ontology matching framework with embedding techniques from multiple views, and design loss functions based on cross-entropy to model different views for learning vector representations of ontological entities.
- We design a novel negative sampling skill tailored for structural relations asserted in ontologies, which can obtain better vector representations of concepts.
- We implement our method and evaluate it on real-world biomedical ontologies. The preliminary result indicates that MultiOM is competitive with several OAEI top-ranked systems in terms of F1-measure.

The rest of this paper is organized as follows. Related work is introduced in Section 2. In Section 3, we introduce the framework of our method in detail. The evaluation of our approach is presented in Section 4, followed by a conclusion with a discussion on future directions in Section 5.

2 Related work

In this section, we review the research efforts on biomedical ontology matching in two aspects as follows.

2.1 Feature-based methods for biomedical ontology matching

There exist various feature-based strategies that are applied on the scenarios biomedical ontology matching, including terminological-based features, structural-based features and employing external semantic thesauruses (e.g., WordNet and UMLS Metathesaurus) as background knowledge for discovering semantically similar entities.

LogMap [4] relies on lexical and structural indexes to enhance its scalability. To scale to large ontologies and minimize the number of logical errors in the aligned ontologies, LogMap uses a horn propositional logic representation of the extended hierarchy of each ontology together with all existing mappings and employs Dowling-Gallier algorithm [20] to model propositional horn satisfiability.

AML [5] is an ontology matching system originally developed to tackle the challenges of matching biomedical ontologies. It employs various sophisticated features and aforementioned domain-specific thesauruses to perform ontology matching. Besides, AML introduces a modularization-based technique to extract the core fragments of the ontologies that contain solely the necessary classes and relations caused by disjoint restrictions. Then, it utilizes confidence-based heuristics to determine near-optimal solutions for incoherent alignments.

FCA-Map [6] is an ontology matching system based on formal concept analysis (FCA), in which five types of formal contexts are constructed in an incremental way, and their derived concept lattices are used to cluster the commonalities among classes and properties at various lexical and structural levels, respectively.

XMap [7] is a scalable matching system that implements parallel processing techniques to enable the composition of basic sophisticated features. It also relies on the employment of external resources such as UMLS Metathesarus to improve the performance of ontology matching.

Recently, CroMatcher [21], as an iterative matching system, is proposed for producing one-to-one final alignment based on ontology structures, which introduces several novelties to the automated weight calculation process. In addition, it applies substitute values for matching modules that are inapplicable for the particular case and utilizes thresholds to eliminate low-probability mapping candidates in alignments. PhenomeNet [22] exploits an axiom-based approach for aligning ontologies, which makes use of the PATO ontology and Entity-Quality definition patterns so as to complement several shortcomings of feature-based methods.

Feature-based methods mainly employ crafting features of the data in order to achieve specific tasks. Unfortunately, determining these hand-crafted features will be limited for a given task and face the bottleneck of improvement. To make matters worse, Cheatham and Hitzler showed that the performance of ontology matching based on such engineered features varies greatly with the domain described by ontologies [23]. As a complement to feature engineering, continuous vectors representing ontological entities can capture the potential associations among features, which is helpful to discover more mappings among ontologies.

2.2 Representation learning methods for biomedical ontology matching

Representation learning techniques have so far limited impacts on ontology matching, specifically in biomedical ontologies. To the best of our knowledge, only four approaches have explored the use of unsupervised representation learning techniques for ontology matching.

Zhang et al. [24] is one of the first that investigate the use of word vectors for ontology matching. They align ontologies based on word2vec vectors [25] trained on Wikipedia. In addition, they use the semantic transformations to complement the lexical information such as names, labels, comments and describing entities. The strategy of entity matching is based on maximum similarity.

Xiang et al. [9] propose an entity representation learning algorithm based on Stacked Auto-Encoders, called ERSOM. To describe an ontological entity (i.e., concept, property), They design a combination of its ID, labels, comments, structural relations and related individuals. The similarity of entities is computed with a fixed point algorithm. Finally, ERSOM generates an alignment based on the stable marriage strategy [26].

DeepAlignment [10] is an unsupervised matching system, which refines pre-trained word vectors aiming at deriving the descriptions of entities for OM. To represent the ontological entities better, the authors represent words by learning their representations and using synonymy and antonymy constraints extracted from general lexical resources and information captured implicitly in ontologies.

SCBOW + DAE(O) [11] is representation learning framework based on terminological embeddings, in which the retrofitted word vectors are introduced and learned by the domain knowledge encoded in ontologies and semantic lexicons. In addition, SCBOW + DAE(O) incorporates an outlier detection mechanism based on a denoising autoencoder that is shown to improve the performance of alignments.

Wang et al. [12] propose a neural architecture tailored for biomedical ontology matching called OntoEmma, It 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.

However, there exist two limitations for above methods. One is the sparsity problem of structural relations. To avoid the poor capability of encoding sparse relations, above methods prefer terminological-based features to learn word vectors for discovering mappings, but they do not make full use of structural relations in ontologies. The other is that these methods rely heavily on the performance of pre-training, which may limit their scalability if there exist no dictionaries or sufficient textual corpora.

3 Muti-view Embedding for Biomedical Ontology Matching

3.1 Problem Statement

To alleviate the heterogeneity of domain ontologies, ontology matching is an effective technique that identifies mappings across ontologies. Before proceed with the presentation of our method, we introduce a formal definition of ontology mapping.

Definition 1 *[1] (Ontology Mapping). Let O_i and O_j be two ontologies. A mapping is a 4-tuple (e_i, e_j, r, n), where e_i and e_j are two entities (i.e., concepts, properties, or individuals) from O_i and O_j, respectively, $r \in \{\sqsubseteq, \sqsupseteq, \equiv\}$ is a relation holding between e_i and e_j, and n is a weight in range $[0, 1]$.*

In the scenario of biomedical ontology matching, matching systems mainly focus on mappings of concepts with equivalent relations. Thus, in the remainder of the paper, we only consider these type of mapping for biomedical ontology matching.

3.2 MultiOM

Existing works [16–18] show that identifying multiple views that can sufficiently represent the data and improve the accuracy and robustness of corresponding tasks. Inspired by their works, we characterize the process of OM from multiple views and try to alleviate the sparsity problem for embedding learning indirectly.

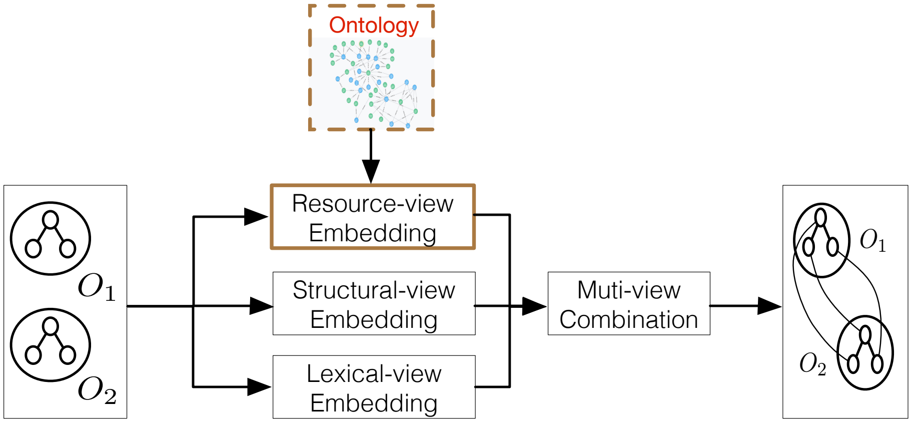

Fig. 1: The framework of MultiOM

The framework of MutiOM is shown in Fig. 1. Given two biomedical ontologies O_1 and O_2, we first extract the concepts and their information of ontologies (e.g., labels, synonyms, structural relations and individuals). Then, we divide the process of ontology matching into three embedding modules from different views, which compose of lexical-view embedding, structural-view embedding and resource-view embedding. Domain ontologies are treated as external resources in the resource-based module, which are treated as bridges to connect source ontology and target one for discovering more potential mappings. With a designed combination strategy based on mutual assessment, we obtain a final alignment among given ontologies.

Different from feature-based methods, we employ extracted information to learn the continuous vector representations of concepts with embedding techniques, which could discover more potential mappings among ontologies. There exist different granularity of vector representations of modules in MultiOM. In lexical-based module, each concept is divided into several tokens $\{t_1, t_2, ..., t_n\}$ that are represented as k-dimensional continuous vectors $t_i, t_j \in \mathbb{R}^k$. The similarity of concepts is measured based on these word vectors by the designed algorithm. Relatively, for structural-based module and resource-based module, each concept C is represented as a d-dimensional continuous vector $\mathbf{C} \in \mathbb{R}^d$, and their similarities are calculated based on cosine measure.

Lexical-view Embedding The lexical-view embedding module is mainly based on TF-IDF algorithm, which is one of the most effective string similarity metrics for ontology matching [23]. According to the assumption of TF-IDF, concepts in one ontology can be represented as a bag of tokens. Then, every concept C_i is regarded as a document and the tokens $\{t_1, t_2, ..., t_l\}$ of each concept are treated as terms. Inspired by the idea soft TF-IDF [23], we propose an embedding-based TF-IDF strategy to calculate the similarities of concepts, More precisely, the similarity of each concept pair is calculated according to the similarities of their tokens, which is obtained based on the cosine measure of tokens' vectors representations rather than the string equivalent of them. The corresponding formula is defined as follows.

$$Sim(C_1, C_2) = \sum_{i=1} w_i \cdot \arg\max_{j} cos(\mathbf{t_{1i}}, \mathbf{t_{2j}}), \tag{1}$$

where C_1 and C_2 are concepts from ontologies O_1 and O_2, $\mathbf{t_{1i}}$ and $\mathbf{t_{2j}}$ are two vector representations of tokens t_{1i} and t_{2i} that belong to C_1 and C_2. w_i is a weight of token $\mathbf{t_{1i}}$ in C_1 that is calculated as follows.

$$w_i = \frac{\text{TFIDF}(t_{1i})}{\sum\limits_{l=1}^{n} \text{TFIDF}(t_{1l})}, \tag{2}$$

where n is the number of tokens, TFIDF($\cdot$) returns the TF-IDF value of each token.

As cosine measure of $\mathbf{t_{1i}}$ and $\mathbf{t_{2j}}$ is a continuous value, so this embedding-based TF-IDF strategy is able to obtain more similar concepts and discover more potential mappings. Nevertheless, our softened strategy depends on the quality of embedding of tokens and may generate more wrong mappings. Therefore, we utilize pre-training vectors to cover the tokens of ontologies as soon as possible (see Section 4.2). On the other hand, we employ the mappings generated by other embedding modules to assess the quality of mappings in lexical-view module (see Section 3.3).

Structural-view Embedding As mentioned before, most proposed methods focus on the terminological-based features to learn word vectors for ontology matching, but they do not make full use of structural relations in ontologies. Relatively, we try to generate mappings from the structural view. To obtain more candidate mappings for training embedding of concepts, we assume that the mappings generated by equivalent strings or their synonym labels are correct, and define a loss function based on cross-entropy to optimize the vector representations of concepts. The loss function is defined as follows.

$$l_{SE} = -\sum_{(C_1,C_2,\equiv,1.0)\in\mathcal{M}} log f_{SE}(C_1,C_2) - \sum_{(C_1',C_2',\equiv,1.0)\in\mathcal{M}'} log(1-f_{SE}(C_1',C_2')), \tag{3}$$

where $\mathcal{M}$ is a set of candidate mappings $\{(C_1,C_2,\equiv,1.0)\}$ generated by our assumption, $\mathcal{M}'$ is a set of negative mappings. We employ the negative sampling skill [13] to generate $\mathcal{M}'$ for training the loss function. For each mapping $(C_i,C_j,\equiv,1.0)\in\mathcal{M}$, we corrupt it and randomly replace C_i or C_j to generate a negative triple $(C_i{}',C_j,\equiv,1.0)$ or $(C_i,C_j{}',\equiv,1.0)$. $f_{SE}(C_1,C_2)$ is a score function defined in Eqs. 4 to calculate the score of concept pairs, where $\mathbf{C_1},\mathbf{C_2}\in\mathbb{R}^d$ are d-dimensional continuous vectors of concepts C_1 and C_2 from different ontologies, $||\cdot||_2$ is the L_2-norm. We hope that $f_{SE}(C_1,C_2)$ is large if concepts C_1 and C_2 are similar.

$$f_{SE}(C_1,C_2) = 2\cdot\frac{1}{1+e^{(||\mathbf{C_1}-\mathbf{C_2}||_2)}}. \tag{4}$$

Furthermore, we design a negative sampling skill tailored for structural relations asserted in ontologies (e.g., *subclassOf* relations, *disjointWith*) relations. Unlike the uniform negative sampling method that samples its replacer from all the concepts, we limit the sampling scope to a group of candidates. More precisely, for each mapping $(C_i,C_j,\equiv,1.0)\in\mathcal{M}$, if there exist *subclassOf* relations (e.g., $(C_i{}',\textit{subclassOf},C_i)$ or $(C_j{}',\textit{subclassOf},C_j)$) asserted in ontologies, we need to exclude this replace case. Relatively, for *disjointWith* relations (e.g.,$(C_i{}',\textit{disjointWith},C_i)$ or $(C_j,\textit{disjointWith},C_j{}')$), we need to give the highest priority to replace cases. With these constrains for negative sampling, we can obtain better vector representations of concepts for OM.

Resource-view Embedding Inspired by the work in [19], we consider external ontology as a bridge to connect two concepts from source ontology and target one. We observe that there exist many different yet overlapping biomedical ontologies such as MA—NCI—FMA, FMA—NCI—SNOMED-CT. Compared with textual descriptions or thesaurus, ontologies as external resources can provide some structural assertions, which is helpful to refine the quality of discovered mappings [19]. Nevertheless, the original idea is mainly based on string equality, which may not discover more similar concepts. Therefore, we employ embedding techniques to soft this framework to discover more potential mappings from this view.

Fig. 2 shows a change of the framework from string equality to the softened idea, where every concept C is represented as a d-dimensional continuous vector $\mathbf{C}\in\mathbb{R}^d$. We assume that there exist some concept pairs (C_1,C_2) involving their synonyms from ontologies O_1 and O_2 will share the same concept C_3 or its synonyms in external ontology O_3. The tuple is labeled as (C_1,C_2,C_3). Then, we introduce two matrices and train them based on these tuples in order to obtain more potential mappings. The loss function is defined as follows.

$$l_{RE} = -\sum_{(C_1,C_2,C_3)\in\mathcal{T}} log f_{RE}(C_1,C_2,C_3) - \sum_{(C_1',C_2',C_3)\in\mathcal{T}'} log(1-f_{RE}(C_1',C_2',C_3)), \tag{5}$$

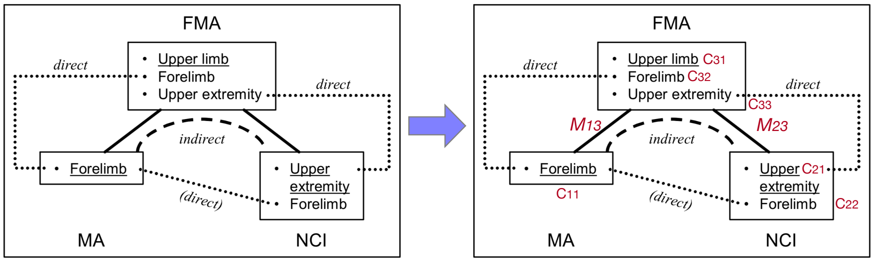

Fig. 2: Left: The original framework for employing external ontology to connect concepts. Right: The embedding framework for employing external ontology to connect concepts

where $\mathcal{T}$ is a set of tuples $\{(C_1, C_2, C_3)\}$ generated by the shared assumption, $\mathcal{T}'$ is a set of negative tuples that randomly replace C_1 or C_2. $f_{RE}(C_1, C_2, C_3)$ is a score function defined in Eqs. 6 to calculate the score of projected concepts, where $\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3} \in \mathbb{R}^d$ are d-dimensional continuous vectors of concepts C_1, C_2, C_3 from different ontologies, M_{13} and M_{23} are two matrices that project $\mathbf{C_1}$, $\mathbf{C_2}$ into the embedding space of O_3, respectively. We hope that the similar concepts will be projected near their shared concept. Conversely, there existed a semantic distance between dissimilar ones.

$$f_{RE}(C_1, C_2, C_3) = 2 \cdot \frac{1}{1 + e^{(||\mathbf{C_1} * M_{13} - \mathbf{C_3}||_2 + ||\mathbf{C_2} * M_{23} - \mathbf{C_3}||_2)}}. \tag{6}$$

To train two matrices better, we maintain all the vectors of concepts in O_3 unchanged and only update the parameters of matrices and concepts from O_1 and O_2. Furthermore, we take advantage of structural relations in O_3 to pre-train the vector representations of concepts, which can adjust semantic distances of concept vectors. As existing KG embedding models face the sparsity problem, we design a loss function based on cross-entropy to achieve this goal that is defined as follows.

$$l_{PT} = -\sum_{(C_{31}, r, C_{32}) \in \mathcal{R}} log f_r(C_{31}, C_{32}) - \sum_{(C'_{31}, r, C'_{32}) \in \mathcal{R}'} log(1 - f_r(C'_{31}, C'_{32})), \tag{7}$$

$$f_r(C_{31}, C_{32}) = 2 \cdot \frac{1}{1 + e^{(||\mathbf{C_{31}} - \mathbf{C_{32}}||_2 - \alpha)}}, \tag{8}$$

where $\mathcal{R}$ is a set of relation assertions, involving $\{(C_{31}, \textit{subClassOf}, C_{32})\} \cup (C_{31}, \textit{PartOf}, C_{32})\}$, $\mathcal{R}'$ is a set of negative ones that randomly replace C_{31} or C_{32}. $f_r(C_{31}, C_{32})$ is a score function that measures the score of (C_{31}, r, C_{32}), $\mathbf{C_{31}}$ and $\mathbf{C_{32}}$ are vector representations of concepts C_{31} and C_{32}. Notice that, *subClassOf* and *PartOf* are not equivalent relations, so we utilize a hyper-parameter α to controls the semantic distances of concept vectors.

3.3 View-Embedding Combination

After obtained mappings from different modules, we need to combine them together. A straightforward strategy is collecting all the mappings from these modules and filtering out them with one threshold or stable marriage algorithm [26]. Although this strategy can obtain a high recall in the final alignment, it may also introduce lots of wrong mappings and miss n:m cases about mappings. Therefore, we propose a combination strategy based on mutual assessment.

For convenience, we use OM-L, OM-S, OM-R to represent the alignments generated by lexical-based module, structural-based module, resource-based module, respectively. The concrete procedures are achieved as follows.

Step 1 Merge the mappings from OM-S and OM-R. Their merged result is labeled as OM-SR, in which the similarity of each mapping is selected the large one between OM-S and OM-R.

Step 2 Select the "reliable" mappings of OM-L and OM-SR based on the corresponding thresholds δ_1 and δ_2.

Step 3 Assess these "reliable" mappings from OM-L and OM-SR mutually. For example, if one "reliable" mapping belongs to OM-L and its similarity in OM-SR is lower than threshold δ_3, then we need to remove it. Relatively, if one "reliable" mapping belongs to OM-SR and its similarity in OM-L is lower than threshold δ_4, then this mapping will be removed.

Step 4 Merge assessed mappings from OM-L and OM-SR and generate a final alignment. For each mapping appearing in OM-L and OM-SR at the same time, its similarity is selected the large one.

4 Experiments

To verify the effectiveness of MultiOM, we used Python to implement our approach with the aid of TensorFlow[10]—a very popular open-source software library for numerical computation. The information of ontologies is parsed by OWLAPI[11], a tool for managing OWL ontologies. The experiments were conducted on a personal workstation with an Intel Xeon E5-2630 V4 CPU which has 64GB memory and TiTAN XP GPU. Our approach[12] can be downloaded together with the datasets and results.

4.1 Datasets

We collect the biomedical ontologies from Anatomy Track in OAEI[13] (Ontology Alignment Evaluation Initiative), which is an annual campaign for evaluating ontology matching systems that attracts many participants all over the world. Furthermore, this campaign provides uniform test cases and standard alignments for measuring precision, recall and F1-measure for all participating systems.

4.2 Experiment Settings

We select several strategies to construct the baseline systems to verify the effectiveness of our model. The following is the detail construction of strategies in our experiments.

- StringEquiv: It is a string matcher based on string equality applied on local names of entities.

[10] https://www.tensorflow.org/

[11] http://owlapi.sourceforge.net/

[12] https://github.com/chunyedxx/MultiOM

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

- StringEquiv + Normalization (StringEquiv-N): It employs normalization techniques[14] before execute StringEquiv matcher.
- StringEquiv + Synonym (StringEquiv-S): It extends the synonym of concepts when executing the StringEquiv matcher[15].
- StringEquiv + Synonym + Reference Ontology (StringEquiv-SR): It introduces external ontologies as bridges to connect concepts based on StringEquiv-S.
- StringEquiv + Synonym + Normalization (StringEquiv-NS): It extends the synonym of concepts when executing the StringEquiv-N.
- StringEquiv + Normalization+ Synonym + Reference Ontology (StringEquiv-NSR): employs normalization techniques before execute StringEquiv-SR.

For MultiOM, we use stochastic gradient descent (SGD) as an optimizer and the configuration of hyper-parameters is listed below: dimensions of concepts and matrices are set to d=$\{\mathbf{50}, 100\}$ and $d_{\mathcal{M}}$=$\{\mathbf{50}, 100\}$. the mini-batch size of SGD is set to Nbatch=$\{5, \mathbf{10}, 20, 50\}$. We select the learning rate λ among $\{\mathbf{0.01}, 0.02, 0.001\}$ and $\{1, 3, \mathbf{5}, 10\}$ negative triples sampled for each positive triple. The whole training spent 1000 epochs. In lexical-based module, the vector presentations of tokens mainly come from the linkage[16] of the work [11], whose dimension is set to **200**. For some tokens without vector presentations, we initialize them randomly and enforce constrains as $||\mathbf{t_{1i}}||_2 \leq 1$ and $||\mathbf{t_{2j}}||_2 \leq 1$. In resource-view embedding module, we employ TransE [13], ConvE [27] and pre-training function 7 to initialize the vector representations of concepts in the external ontology. α is set to $\{\mathbf{0.01}, 0.05, 0.10\}$ in loss function 7 so as to control the semantic distances of concept vectors. For negative sampling strategy, we collect all the related structural assertions of concepts. When one concept was selected as a replacer, we retrieve the structural assertions of this concept and execute the replacement based on its relations with the original concept. During this process of replacement, *disjointWith* relations[17] own the highest priority and *subclassOf* relations should be excluded. Finally, the result of MultiOM is generated by the combination strategy, and we set the related thresholds $\delta_1 = 0.8$, $\delta_2 = 0.95$, $\delta_3 = 0.65$, $\delta_4 = 0.3$.

In order to show the effect of our proposed negative sampling, a symbol "-" added to module labels or merge ones indicates that this module is not equipped with negative sampling tailored for structural relations.

4.3 The evaluation results

Table 1 lists the matching results of MultiOM compared with baseline systems. We observe that merging more strategies can improve the number of mappings. Although it slightly decreases the precision of alignments, it can increase the recall and F1-measure as a whole. Relatively, MultiOM further improves the recall and F1-measure of alignments because continue vector representations of concepts can obtain more similar concepts and discover more potential mappings. Moreover, the performance of MultiOM is better than MultiOM^{-} in term of F1-measure. The main reason is that employing structural relations are helpful to distinguish the vector representations of concepts.

[14] https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/2013/docs/userDoc/tools/norm.html

[15] For biomedical ontologies, there exists at least one synonym of each concept in most cases.

[16] https://doi.org/10.5281/zenodo.1173936.

[17] For large biomedical ontologies, extracted *disjointWith* relations are only referred to the concepts and their children.

Table 1: The comparison of MultiOM with baseline systems

Methods	MA-NCI					FMA-NCI-small				
	Number	Correct	P	R	F1	Number	Correct	P	R	F1
StringEquiv	935	932	0.997	0.615	0.761	1501	1389	0.995	0.517	0.681
StringEquiv-N	992	989	0.997	0.625	0.789	1716	1598	0.995	0.595	0.863
StringEquiv-S	1100	1057	0.961	0.697	0.808	2343	2082	0.974	0.775	745
StringEquiv-SR	1162	1094	0.941	0.722	0.817	2343	2082	0.974	0.775	745
StringEquiv-NS	1153	1109	0.962	0.732	0.831	2464	2200	0.975	0.819	0.890
StringEquiv-NSR	1211	1143	0.943	0.753	0.838	2464	2200	0.975	0.819	0.890
MultiOM^{-}	1484	1296	0.873	0.855	0.864	2500	2173	0.947	0.809	0.872
MultiOM	1445	1287	0.891	0.849	0.869	2538	2195	0.942	0.817	0.875

Table 2: The results about combining with different embedding modules in Anatomy Track

Methods	Number	Correct	P	R	F1
TFIDF (threshold$= 0.8$)	985	976	0.991	0.644	0.780
OM-L (threshold$= 0.8$)	1286	1175	0.914	0.775	0.839
OM-S^{-} (threshold$= 0.95$)	1836	1109	0.604	0.732	0.662
OM-S (threshold$= 0.95$)	1189	1097	0.923	0.724	0.811
OM-R (Random initialization, threshold$= 0.95$)	709	680	0.959	0.449	0.661
OM-R (TransE, threshold$= 0.95$)	22	4	0.182	0.003	0.005
OM-R (ConvE, threshold$= 0.95$)	835	790	0.946	0.521	0.672
OM-R (loss function 7, threshold$= 0.95$)	833	789	0.948	0.520	0.672
OM-RS^{-} (threshold$= 0.95$)	1271	1147	0.902	0.757	0.823
OM-RS (threshold$= 0.95$)	1237	1138	0.920	0.751	0.827
MultiOM^{-}	1484	1296	0.873	0.855	0.864
MultiOM	1445	1287	0.891	0.849	0.869

Table 2 shows the comparison of different combination with embedding-view modules. Overall, merge more embedding-view modules, the performances of alignments are better. For lexical-view module, softened TF-IDF (denoted as OM-L) is better than original TF-IDF in terms of F1-measure because continuous vectors representing tokens can provide more semantic information than single strings for calculating the similarity of concepts. For resource-view embedding module (denoted as OM-R), ConvE and our pre-training function are better than random initialization because both of them can utilize structural relations to adopt vector representations of concepts in the semantic space. Nevertheless, compared with 20 minutes spent in function 7, ConvE took nearly 24 hours to obtain the vector presentations of concepts. Notice that, it is not suitable for TransE to pre-train the vector presentations of concepts. We analyze that sparse structural relations of ontologies and its simplified score function limit its capability. Overall, we observe that employing new negative sampling strategy in embedding-view modules (i.e., OM-S, OM-RS[18], MultiOM) is helpful to improve the quality of alignments further in terms of precision and F1-measure.

Table 3 indicates that different embedding-view modules are complementary to each other. As OM-L can obtain more mappings than others, it can discover more correct mappings than OM-S, OM-R and their merged case.

Table 3: The external correct mappings discovered in each embedding-view module

	OM-L	OM-S	OM-R	OM-LS	OM-LR	OM-SR
OM-L	–	176	463	–	–	154
OM-S	99	–	354	–	45	–
OM-R	78	46	–	24	–	–

Table 4 lists the comparison of MultiOM with OAEI 2018 top-ranked systems based on feature engineering and SCBOW + DAE(O) based on representation learning. It shows that the preliminary result of MultiOM can be competitive with several promising matching systems (e,g, FCAMapX and SANOM) in terms of F1-measure. Nevertheless, there still exists a gap compared with the best systems (e.g., AML and SCBOW + DAE (O)). We analyze that lexical-based module and simplified combination strategy may become the main bottleneck of MultiOM. Benefited from thesauruses (e.g., UMLS) and optimized combination strategy, most top-ranked systems can obtain better performances of OM tasks. In addition, most systems (e.g., AML, LogMap) employ alignment debugging techniques, which is helpful to improve the quality of alignment further. But we do not employ these techniques in the current version. We leave these issues in our future work.

5 Conclusion and future work

In this paper, we presented an alternative OM framework based on multi-view embedding techniques, in which different loss functions were designed based on cross-

[18] OM-RS is achieved according to Step 1 in our combination strategy, and we choose threshold (δ=0.95) to obtain the best F1-measure.

[19] There exists a slightly difference of the result of SCBOW + DAE(O) because the number of reference mappings is 1516 rather than 1497.

Table 4: The comparison of MultiOM with OAEI 2018 top-ranked systems

Methods	MA-NCI					FMA-NCI-small				
	Number	Correct	P	R	F1	Number	Correct	P	R	F1
AML	1493	1418	0.95	0.936	0.943	2723	2608	0.958	0.910	0.933
SCBOW + DAE(O)[19]	1399	1356	0.969	0.906	0.938	2282	2227	0.976	0.889	0.930
LogMapBio	1550	1376	0.888	0.908	0.898	2776	2632	0.948	0.902	0.921
POMAP++	1446	1329	0.919	0.877	0.897	2414	2363	0.979	0.814	0.889
XMap	1413	1312	0.929	0.865	0.896	2315	2262	0.977	0.783	0.869
LogMap	1387	1273	0.918	0.846	0.880	2747	2593	0.944	0.897	0.920
SANOM	1450	1287	0.888	0.844	0.865	–	–	–	–	–
FCAMapX	1274	1199	0.941	0.791	0.859	2828	2681	0.948	0.911	0.929
MultiOM	**1445**	**1287**	**0.891**	**0.849**	**0.869**	**2538**	**2195**	**0.942**	**0.817**	**0.875**

entropy to model different view among ontologies and learn the vector representations of concepts. We further proposed a novel negative sampling skill tailored for structural relations asserted in ontologies, which could obtain better vector representations of concepts. We implemented our method and evaluated it on real-world biomedical ontologies. The preliminary result indicated that MultiOM was competitive with several OAEI top-ranked systems in terms of F1-measure.

In the future work, we will explore the following research directions: (1) As the candidate mappings and tuples are not enough, we will extend MultiOM to an iterative framework. (2) The combination strategy in MultiOM is too subjective. Recently, Zhang et al. [18] presented combination strategies for entity alignment based on embedding techniques. Incorporating these strategies into MultiOM may facilitate improving the quality of mappings. (3) Some senior symbolic reasoning techniques (e.g., incoherent checking) could be served for training process and alignment generation. We will merge them into MultiOM for improving its performances.

References

1. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Science & Business Media, Heidelberg, 2013.
2. Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching: A literature review. *Expert Syst. Appl.*, 42(2):949–971, 2015.
3. Ian Harrow, Ernesto Jiménez-Ruiz, Andrea Splendiani, Martin Romacker, Peter Woollard, Scott Markel, Yasmin Alam-Faruque, Martin Koch, James Malone, and Arild Waaler. Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 8(1):1–13, 2017.
4. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-Based and Scalable Ontology Matching. In *Proceedings of ISWC*, pages 273–288, 2011.
5. Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F Cruz, and Francisco M Couto. The AgreementMakerLight Ontology Matching System. In *Proceedings of OTM Conferences*, pages 527–541, 2013.
6. Mengyi Zhao, Songmao Zhang, Weizhuo Li, and Guowei Chen. Matching biomedical ontologies based on formal concept analysis. *Journal of Biomedical Semantics*, 9(1):11, 2018.
7. Warith Eddine Djeddi and Mohamed Tarek Khadir. A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies. In *Proceedings of Data Warehousing and Knowledge Discovery*, pages 320–331, 2014.

8. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
9. Chuncheng Xiang, Tingsong Jiang, Baobao Chang, and Zhifang Sui. ERSOM: A Structural Ontology Matching Approach Using Automatically Learned Entity Representation. In *Proceedings of EMNLP*, pages 2419–2429, 2015.
10. Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In *Proceedings of NAACL*, pages 787–798, 2018.
11. Prodromos Kolyvakis, Alexandros Kalousis, Barry Smith, and Dimitris Kiritsis. Biomedical ontology alignment: an approach based on representation learning. *Journal of Biomedical Semantics*, 9(1):21, 2018.
12. Lucy Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, and Waleed Ammar. Ontology alignment in the biomedical domain using entity definitions and context. In *Proceedings of BioNLP*, pages 47–55, 2018.
13. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of NeurIPS*, pages 2787–2795, 2013.
14. Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *Proceedings of WWW*, pages 2366–2377, 2019.
15. George A Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
16. Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. An attention-based collaboration framework for multi-view network representation learning. In *CIKM*, pages 1767–1776, 2017.
17. Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. Semi-supervised sequence modeling with cross-view training. In *EMNLP*, pages 1914–1925, 2018.
18. Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Multi-view Knowledge Graph Embedding for Entity Alignment. In *Proceedings of IJCAI*, 2019.
19. Songmao Zhang and Olivier Bodenreider. Experience in Aligning Anatomical Ontologies. *International Journal on Semantic Web and Information Systems*, 3(2):1–26, 2007.
20. Maria Grazia Scutella. A Note on Dowling and Gallier's Top-Down Algorithm for Propositional Horn Satisfiability. *Journal of Logic Programming*, 8(3):265–273, 1990.
21. Marko Gulić, Boris Vrdoljak, and Marko Banek. Cromatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. *Journal of Web Semantics*, 41:50–71, 2016.
22. Miguel Ángel Rodríguez-García, Georgios V Gkoutos, Paul N Schofield, and Robert Hoehndorf. Integrating phenotype ontologies with PhenomeNET. *Journal of Biomedical Semantics*, 8(1):58, 2017.
23. Michelle Cheatham and Pascal Hitzler. String Similarity Metrics for Ontology Alignment. In *Proceedings of ISWC*, pages 294–309, 2013.
24. Yuanzhe Zhang, Xuepeng Wang, Siwei Lai, Shizhu He, Kang Liu, Jun Zhao, and Xueqiang Lv. Ontology Matching with Word Embeddings. In *Proceedings of CCL*, pages 34–45. 2014.
25. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NeurIPS*, pages 3111–3119, 2013.
26. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of ICDE*, pages 117–128, 2002.
27. Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of AAAI*, pages 1811–1818, 2018.