Exploring Wasserstein Distance across Concept Embeddings for Ontology Matching

Yuan An, Alex Kalinowski, and Jane Greenberg

College of Computing and Informatics, Drexel University, Philadelphia, PA, USA
{ya45,ajk437,jg3243}@drexel.edu

Abstract. Measuring the distance between ontological elements is fundamental for ontology matching. String-based distance metrics are notorious for shallow syntactic matching. In this exploratory study, we investigate Wasserstein distance targeting continuous space that can incorporate various types of information. We use a pre-trained word embeddings system to embed ontology element labels. We examine the effectiveness of Wasserstein distance for measuring similarity between ontologies, and discovering and refining matchings between individual elements. Our experiments with the OAEI conference track and MSE benchmarks achieved competitive results compared to the leading systems.

Keywords: ontology matching · optimal transport · Wasserstein Distance · ontology embedding.

1 Introduction

Semantic and structural heterogeneity is widespread among ontologies. To bridge the heterogeneity, almost all ontology matching systems [24, 11, 17] need to evaluate the distances between ontological elements. String-based distance metrics have dominated the field [6]. However, string-based distance metrics are notorious for shallow syntactic matching. It is also challenging to determine which string similarity measures to use and how to effectively combine them in matching systems [6]. Aligning elements across different ontologies essentially involves measuring the semantic similarity/distance of the ontological elements representing items in the underlying domains. Recently, word embeddings [22] have been used to successfully encode syntactic and semantic word relationships. As a result, word embeddings have displayed excellent performance in applications for cross-lingual word alignment [14]. Ontology matching techniques have also been created using embeddings, with embedding vectors predominantly utilized as inputs in supervised [23, 16] or distantly supervised [7] machine learning models.

While machine learning is effective in making use of ontology embeddings, a significant effort is required to gather training instances. Although some systems directly use the cosine similarity between embedding vectors [18] for deriving candidate matchings, the embeddings are first retrofitted through tailored training instances. In practice, the present top matching algorithms are largely unsupervised and rely only upon existing ontology and external sources. Motivated by

the desired property of unsupervised learning, we posit the following question: *If ontology elements can be readily encoded as embedding vectors, can we use them in an unsupervised fashion for ontology matching?*

To research the question, we formulate ontology matching as an optimal transport problem from a source ontology embedding space to a target ontology embedding space. Optimal Transport (OT) [27] has been applied to various alignment applications including word embeddings alignment [12], sequence-to-sequence learning [9], heterogeneous domain alignment [8], and graph comparison and matching [21]. A desired advantage of OT-based approach is *unsupervised learning*. The OT solution establishes optimal mappings and a shape-based distance called *Wasserstein distance* between distributions. In this study, we explore the effectiveness of Wasserstein distance for ontology matching. Our inquiry focuses on answering the following 3 questions:

1. *How effective is Wasserstein distance for measuring similarity between (blocks of) ontologies?*
2. *How effetive is the coupling matrix accompanying a Wasserstain distance for deriving alignments between individual ontological elements?*
3. *How effective is Wasserstain distance for refining matching candidates?*

The rest of the paper presents our research and is organized as follows. Section 2 introduces optimal transport and Wasswserstein distance. Section 3 describes the ontology embeddings used in this study. Section 4 measures ontology similarity in Wasswerstein distance. Section 5 derives matching candidates. Section 6 refines matching candidates. Section 7 presents our experiment and results. Section 8 comments on the results. Section 9 discusses related work, and finally, Section 10 concludes the paper with future directions.

2   Optimal Transport and Wasserstein Distance

Optimal transport (OT) [27] originated as a solution to the problem of transporting masses from one configuration onto another with *the least effort*. OT had deep connections with major economic problems, for example, in logistics, production planning, and network routing, etc. Since then, optimal transport has been generalized from practical concerns to powerful mathematical tools for comparing distributions. In particular, given two point sets modeled as two discrete distributions, optimal transport is an effective approach for discovering a minimum cost mapping between the two sets. Considering the set of embedding vectors of an ontology as a set of points, we can apply optimal transport to discover a minimum cost mapping between two sets of ontology embeddings.

Formally, given two sets of ontology embeddings $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d, i = 1..n\}$ and $\mathbf{Y} = \{\mathbf{y}_j \in \mathbb{R}^d, j = 1..m\}$, where each embedding is represented as a vector $\mathbf{x}_i$ or $\mathbf{y}_j \in \mathbb{R}^d$. Let $\mu = \sum_{i=1}^{n} p(\mathbf{x}_i)\delta_{\mathbf{x}_i}$ be the probability distribution defined on the set $\mathbf{X}$, where $\delta_{\mathbf{x}_i}$ is the Dirac at the point $\mathbf{x}_i$ and $p(\mathbf{x}_i)$ is a probability weight associated with the point $\mathbf{x}_i$. Similarly, let $\nu = \sum_{j=1}^{n} q(\mathbf{y}_j)\delta_{\mathbf{y}_j}$ be the probability distribution defined on the set $\mathbf{Y}$, where $\delta_{\mathbf{y}_j}$ is the Dirac at the point $\mathbf{y}_j$ and $q(\mathbf{y}_j)$ is a probability weight associated with the point $\mathbf{y}_j$. Usually, we consider uniform weights, e.g., $p(\mathbf{x}_i) = \frac{1}{n}$, for $i = 1..n$, and $q(\mathbf{y}_j) = \frac{1}{m}$,

for $j = 1..m$. However, if additional information is provided, $p(\mathbf{x}_i)$ and $q(\mathbf{y}_j)$ can incorporate the information as non-uniform distributions. Optimal transport (OT) defines an optimal plan for mass transportation and a distance between the two distributions.

Specifically, let $\mathbf{C} = [c(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$ be a ground cost matrix with $c(\mathbf{x}_i, \mathbf{y}_j)$ measuring a ground distance between the individual embeddings $\mathbf{x}_i$ and $\mathbf{y}_j$. Let $\mathbf{T} = [T(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$ be a matrix of a transport plan (or *couplings*) with $T(\mathbf{x}_i, \mathbf{y}_j)$ specifying how much mass will be transported from point $\mathbf{x}_i$ to point $\mathbf{y}_j$. Let $\Pi(\mu, \nu)$ be the set of all feasible transport plans defined as:

$$\Pi(\mu, \nu) \stackrel{\text{def}}{=} \{\mathbf{T} \in \mathbb{R}_+^{n \times m} | \mathbf{T}\mathbb{1}_n = \mu, \mathbf{T}^\top \mathbb{1}_m = \nu\} \tag{1}$$

where $\mathbb{1}_n$ and $\mathbb{1}_m$ are all one vectors, $\mathbf{T}\mathbb{1}_n = \mu$ and $\mathbf{T}^\top \mathbb{1}_m = \nu$ are marginal constraints on feasible plans. The *Optimal Transport* problem is to find the map $\mathbf{T} : \mathbf{X} \to \mathbf{Y}$, where

$$\mathbf{T} = \underset{\mathbf{T} \in \Pi(\mu, \nu)}{\text{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{m} c(\mathbf{x}_i, \mathbf{y}_j) \cdot T(\mathbf{x}_i, \mathbf{y}_j), \text{s.t.}, \mathbf{T}\mathbb{1}_n = \mu, \mathbf{T}^\top \mathbb{1}_m = \nu \tag{2}$$

The map $\mathbf{T}$ is also called *coupling matrix*. It gives rise to a distance measure between the two distributions called *Wasserstein distance* defined as:

$$W(\mu, \nu) \stackrel{\text{def}}{=} \min_{\mathbf{T} \in \Pi(\mu, \nu)} \langle \mathbf{C}, \mathbf{T} \rangle = \min_{\mathbf{T} \in \Pi(\mu, \nu)} \sum_{i=1}^{n} \sum_{j=1}^{m} c(\mathbf{x}_i, \mathbf{y}_j) \cdot T(\mathbf{x}_i, \mathbf{y}_j) \tag{3}$$

The optimization problem can be efficiently solved by replacing the objective with an entropy regularized objective such as in the sinkhorn algorithm [27].
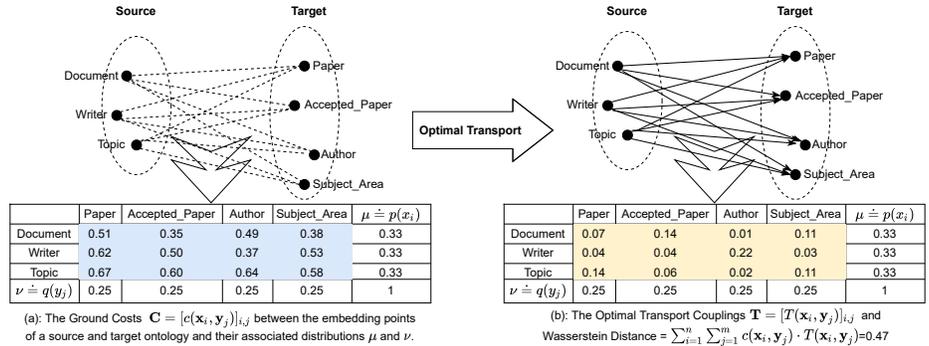

	Paper	Accepted_Paper	Author	Subject_Area	$\mu \stackrel{.}{=} p(x_i)$
Document	0.51	0.35	0.49	0.38	0.33
Writer	0.62	0.50	0.37	0.53	0.33
Topic	0.67	0.60	0.64	0.58	0.33
$\nu \stackrel{.}{=} q(y_j)$	0.25	0.25	0.25	0.25	1

(a): The Ground Costs $\mathbf{C} = [c(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$ between the embedding points of a source and target ontology and their associated distributions μ and ν.

	Paper	Accepted_Paper	Author	Subject_Area	$\mu \stackrel{.}{=} p(x_i)$
Document	0.07	0.14	0.01	0.11	0.33
Writer	0.04	0.04	0.22	0.03	0.33
Topic	0.14	0.06	0.02	0.11	0.33
$\nu \stackrel{.}{=} q(y_j)$	0.25	0.25	0.25	0.25	1

(b): The Optimal Transport Couplings $\mathbf{T} = [T(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$ and Wasserstein Distance $= \sum_{i=1}^{n} \sum_{j=1}^{m} c(\mathbf{x}_i, \mathbf{y}_j) \cdot T(\mathbf{x}_i, \mathbf{y}_j) = 0.47$

Fig. 1. Optimal Transport Problem between a Source and Target Ontology

Example 1. Figure 1 illustrates the configuration of the optimal transport problem between a source and target ontology. The source ontology has 3 concepts:

Document, Writer and *Topic*. *The target ontology has 4 concepts which are: Paper, Accepted_Paper, Author and Subject_Area. The table in Figure 1 (a) shows the Euclidean distances as the ground costs between the embeddings of the source and target ontology concepts. The points in both the source and target embedding sets are uniformly distributed, as reported in the last column μ and the last row ν. By solving Equation (2), the table in Figure 1 (b) displays the optimal transport couplings between the points in the source and target embedding spaces. Notice the couplings are feasible because they satisfy the following marginal constraints: $\mathbf{T}\mathbb{1}_n = \mu$ and $\mathbf{T}^\top \mathbb{1}_m = \nu$. By solving the Equation (3), the Wasserstein distance between the two embedding sets is 0.47 given the ground costs and the couplings.*

3   Ontology Embeddings and Matching

Ontology embedding is the problem of encoding ontological elements as numerical vectors. Various methods have been proposed for representing individual components in an ontology as embeddings. For example, translational-based methods [5, 19] and graph neural networks (GNN) [15] encode an ontology based on its graph structure. Text-enhanced methods for ontology embeddings [20, 26, 2, 28, 29] encode lexical words of ontology elements. Logic-aware methods [25, 10] incorporate logical constraints into ontology embeddings. It is attempting to encode ontologies using the above methods and directly apply optimal transport for discovering mapping. However, simply applying these embeddings for ontology matching is negatively impacted by the *lack of registration* problem [1]. Specifically, these embeddings were mainly developed for applications concerned with a single ontology, for example, link prediction [13]. The embedding spaces of two independent ontologies may mismatch due to different dimensions or various rotations and translations. As a result, there may not exist a distance or the direct geometric locations between the points in the embedding spaces may not reflect their underlying genuine relationships. This is a significant issue for optimal transport-based approach which needs a meaningful ground cost.

We will study the problem of matching the embeddings incorporating structural and logical information in future work. In this exploratory study, we focus on the ontology embeddings corresponding to the labels of the ontology elements, for example, the *'rdfs:label'* of a concept. we apply the pre-trained language model, *fasttext* [4], to encode the labels of the set of ontology concepts, object and datatype properties. Using the same pre-trained language model to encode the labels of different ontologies will alleviate the lack of registration problem, because the resultant embeddings are in the same embedding space. We will show that OT on label embeddings already produce promising results. For each element, we first normalizes the element's label via a sequence of standard text processing steps. If necessary, the labels are augmented with synonyms. We then split the normalized label into individual words which in turn are fed into the pre-trained language model to obtain their corresponding word embeddings. For the element, we obtain its embedding by computing the average of the set of embeddings or use the entire set of the embeddings of the individual words for next-step processing.
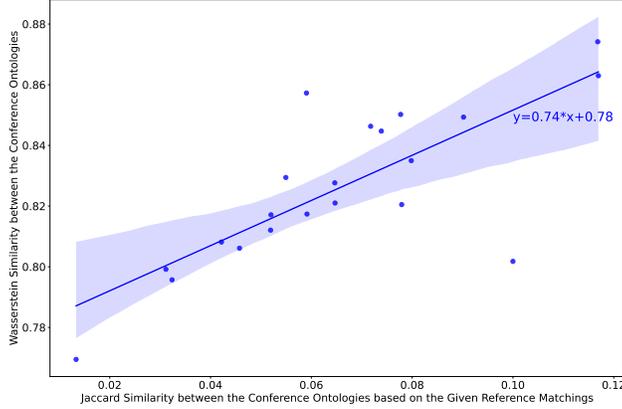
Fig. 2. Regression plot between Wasserstein similarity (e^{-wd}) and Jaccard similarity based on the given matching references for the conference ontologies

4 Wasserstain Distance for Measuring Ontology Similarity

Prior to applying optimal transport and Wasserstein distance for ontology matching, we first analyze some properties of Wasserstein distance for capturing ontology similarity. A desirable property is that Wasserstein distance should closely correlate with ontology similarity. That is, the more similar two ontologies are, the shorter the Wasserstein distance between them is. Quantitatively measuring the similarity between two ontologies is a very challenging problem. One option is to count the minimum number of graph edit operations to transform one graph to another. It is known that graph edit distance is NP-hard and it depends on a set of graph edit operations. Another option is the Jaccard Index on sets. If matchings between ontologies are available, we can define the Jaccard similarity between ontologies as follows:

$$Jac(\mathcal{O}_S, \mathcal{O}_T) = \frac{|M|}{|\mathcal{O}_S| + |\mathcal{O}_T| - |M|},$$

where $\mathcal{O}_S$ and $\mathcal{O}_T$ are source and target ontologies, $|\mathcal{O}_S|$ and $|\mathcal{O}_T|$ return the number of concepts in each ontology, M is the set of matchings between $\mathcal{O}_S$ and $\mathcal{O}_T$, and $|M|$ gives the number of matchings.

The OAEI Campaign provides a list of ontology matching tasks each with a set of curated matching references. In particular, the Conference track contains 21 matching cases among 7 conference ontologies. For each case, we compute: (1) its Jaccard similarity based on the given matching references, and (2) the Wasserstein distance between the ontology embeddings. We convert a Wasserstein distance, wd, to a Wasserstein similarity, ws as $ws = e^{-wd}$. Figure 2 shows the regression plot between the Wasserstein similarities and Jaccard similarities. Furthermore, we calculated the Pearson correlation coefficient (PCC) between the two sets of similarities defined as $\rho = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$. The PCC is 0.77, indicat-

ing Wasserstein distance is highly correlated with the Jaccard coefficients when matchings between two ontologies are known. As a result, Wasserstein distance exhibits the desirable property for capturing ontology similarity. We can then leverage this good property for developing algorithms that derive and refine matchings using optimal transport as presented below.

5 Deriving Matching Candidates from Global Coupling Matrix

Given a source $\mathcal{O}_S$ and a target $\mathcal{O}_T$ ontology, we first apply optimal transport *globally* to the ontologies for generating a set of candidate matchings. We then compute contextual *local* Wasserstein distances to refine the matchings by filtering out false positives.

In this section, we describe the building blocks for deriving matching candidates globally. Let $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d, i = 1..n\}$ be the source ontology concept embeddings and $\mathbf{Y} = \{\mathbf{y}_j \in \mathbb{R}^d, j = 1..m\}$ be the target ontology concept embeddings. The optimal transport problem defined in Equation (2) requires as input ground costs $\mathbf{C} = [c(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$ and probability distributions $p(\mathbf{x}_i)$ and $q(\mathbf{y}_j)$. For label embedding spaces, we use the Euclidean distances as the ground costs. For the probability distributions, we estimate non-uniform weights using the shortest distances between source and target concept embeddings. In particular, for a source point $\mathbf{x}_i \in \mathbf{X}$, let $d_i = \min_{j=1}^m c(\mathbf{x}_i, \mathbf{y}_j)$ be the shortest distance from $\mathbf{x}_i$ to all embedding points in the target space. The distribution $p(\mathbf{x}_i)$ will be inversely proportional to d_i for $i = 1..n$. In other words, the greater the shortest distance from a source point to all target points, the less the weight associated with the source concept. Similarly, we estimate non-uniform probability distribution $q(\mathbf{y}_j)$ associated with the target concepts using the shortest distances from target points to source points.

The solution, $\mathbf{T} = [T(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}, i = 1..n, j = 1..m$, is a coupling matrix between every source and target embedding point. We test the following two methods for deriving candidate matchings from the coupling matrix:

- **Mutual Nearest Neighbor (MNN):** for a $\mathbf{x}_p \in \{\mathbf{x}_i \in \mathbb{R}^d, i = 1..n\}$, find $\mathbf{y}_q \in \{\mathbf{y}_j \in \mathbb{R}^d, j = 1..m\}$, such that, $T(\mathbf{x}_p, \mathbf{y}_q) = \max\{T(\mathbf{x}_p, \mathbf{y}_j), j = 1..m\}$ and $T(\mathbf{x}_p, \mathbf{y}_q) = \max\{T(\mathbf{x}_i, \mathbf{y}_q), i = 1..n\}$.
- **Top-K Targets (TopK):** for a $\mathbf{x}_p \in \{\mathbf{x}_i \in \mathbb{R}^d, i = 1..n\}$, find k targets $\{\mathbf{y}_{q_1}, \mathbf{y}_{q_2}, .., \mathbf{y}_{q_k}\} \subset \{\mathbf{y}_j \in \mathbb{R}^d, j = 1..m\}$, such that, $T(\mathbf{x}_p, \mathbf{y}_{q_z}) \geq \max\{T(\mathbf{x}_p, \mathbf{y}_j), j \neq q_1..q_k\}$, for $z = 1..k$.

Example 2. In Figure 1(a), the shortest distances from the source concepts to the targets concepts are $d_{Document} = 0.35$, $d_{Writer} = 0.37$, and $d_{Topic} = 0.58$. Taking inverses of the distances and normalizing them gives rise to a non-uniform source distribution $\mu = \{p(Document) = 0.39, p(Write) = 0.37, p(Topic) = 0.24\}$. We can estimate the target probability distribution in the same way. After solving the optimal transport problem, we obtained the coupling matrix for the optimal transportation as shown in Figure 1(b). By applying MNN, we obtain the following correspondences: Document⤳Accepted_Paper, Write⤳Author, Topic⤳Paper.

(a) Compute the Wasserstein Distance (WD) betwen each pair of triples using the Euclidean Distances (ED) between the individua elements in the triples as the ground costs

(b) Compute the Wasserstein Distance (WD) betwen the contexs the source concept 'Presentation' and the target concept 'Paper_Presentation' using the WDs between individual triples as the ground costs
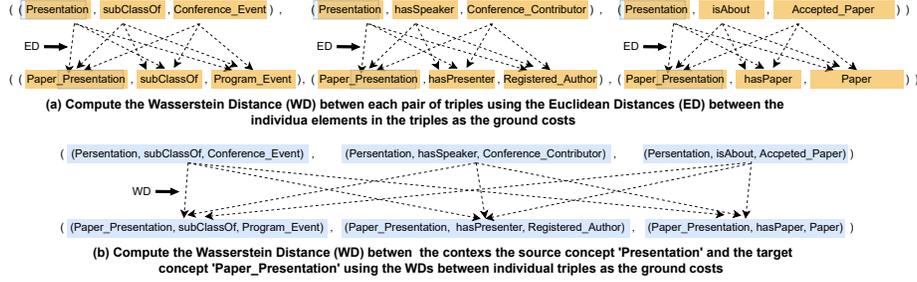
Fig. 3. Illustration of computing local Wasserstein Distance (localWD) between the local contexts of a source concept 'Presentation' and a target concept 'Paper_Presentation'. The computation starts with computing the pairwise Wasserstein distance (pairWD) between each pair of triples in the contexts (a). It then computes the localWD using the pairWDs as the ground costs (b).

By applying TopK (K=2), we obtain a different set of correspondences each of which contains a set of potential target concepts for each source as follows:
Document⤳{Accepted_Paper, Subject_Area},
Write⤳{Author, Paper},
Topic⤳{Paper, Subject_Area}.

Our experimental results (see Section 7) demonstrated that the global matching candidates (through MNN and TopK) outperformed most of the SOTA systems in terms of the *recall* metrics. To improve its overall F1 measures, we develop refinement steps presented in next section.

6   Refining Matching Candidates by Local Wasserstein Distances

We refine the candidate matchings by using Wasserstein distances between the local contexts of source and target ontology elements. The local context of an element contains the triples involving the element, including the subClassOf relationships connecting to the element's parents and children, object property, and datatype property. For each matching candidate, we retrieve the local contexts of the source and target concepts. We then compute the local Wasserstein distance (localWD) between the local contexts by the following steps. First, compute the pairwise Wasserstein distance (pairWD) between each pair of triples in the contexts. The ground costs for computing the pairWDs are the Euclidean distances between the embeddings of the individual elements in the triples. Second, compute the local Wasserstein distance between the contexts by using the pairWDs as ground costs.

Example 3.   Figure 3 illustrates the two-step procedure for computing the localWD for a matching candidate Presentation⤳Paper_Presentation. We first extract the local context of the source concept Presentation as a set of triples, context(Presentation) = {S_1, S_2, S_3}, as follows:
S_1:(Presentation, subClassOf, Conference_Event),

S_2:(Presentation, hasSpeaker, Conference_Contributor),
S_3:(Presentation, isAbout, Accepted_Paper).
Similarly, we extract the local context of the target concept Paper_Presentation *as a set of triples, context(*Paper_Presentation*) = $\{T_1, T_2, T_3\}$, as follows:*
T_1:(Paper_Presentation, subClassOf, Program_Event),
T_2:(Paper_Presentation, hasPresenter, Registered_Author),
T_3:(Paper_Presentation, hasPaper, Paper).
*Given these two local contexts, we compute the pairwise WDs between the two sets of triples as illustrated in Figure 3(a) (where only $S_1 \rightsquigarrow T_1$, $S_2 \rightsquigarrow T_2$, and $S_3 \rightsquigarrow T_3$ are shown). Finally, we use the pairWDs as ground costs to compute the localWD between context(*Presentation*) and context(*Paper_Presentation*) as illustrated in Figure 3(b).*

After computing the local WDs for all matching candidates, we refine the candidate matchings using the localWDs as a main factor. We describe the set of experiments and results in next section.

7 Experiment and Results

Setting Up. We name the process of deriving matching candidates from global coupling matrix as *OTMapOnto*_global. We add a suffix _mnn or _topK to it to indicate whether the candidates are derived by mutual nearest neighbors or top-K targets. In our experiments, we derive a large set of candidates by top-20 targets for refinement (thus, *OTMapOnto*-global_top20 in the following tables illustrating the results). We name the process of refining matching candidates through local Wasserstein distances as *OTMapOnto*_refinement. We evaluate these processes on the Conference track [1] and MSE benchmark[2] in the OAEI Campaign. The code and Jupyter notebooks for the experiments is here[3].

In the refinement process, we create interactions among localWDs, string-based distances, and embedding-based Euclidean distances. The interactions are performed by multiplication. We then examine any enhancements brought by the localWDs in comparison to only string-based distances, embedding-based Euclidean distances, and their interactions. We compare the following cases:

- String-based Levenshtein distances/similarities (string-based distance)
- Interactions between the string-based distances and localWDs (string-context-distance)
- Euclidean distances between the averaged embeddings of labels
- Interactions between the Euclidean distances and localWDs
- Wasserstein distances between the embeddings of labels
- Interactions between the label Wasserstein distances and localWDs
- Interactions among string-based distances, Euclidean distances, label Wasserstein distances, and localWDs

We converted each distance metric x, to a similarity metric in $[0, 1]$ by taking e^{-x}. We run through thresholds from 0 to 1 in a step of 0.01 to find the best

---

[1] http://oaei.ontologymatching.org/2021/conference/index.html
[2] https://github.com/EngyNasr/MSE-Benchmark
[3] https://github.com/anyuanay/otmaponto_django

performance. Our experimental results show that the interactions between the string-based distances and localWDs (*string-context-distance*) achieve the best performance among all distance metrics. In the following tables, we specifically report on the values related to the *string-context-distance* metric.

Evaluation on Conference Track. There are 21 matching cases among 7 conference ontologies. We adopt the main reference alignment **rar2** with class only case (**M1**) for evaluation. Table 1 contains the **rar2-M1** results of the OAEI 2021 Campaign, plus the **rar2-M1** results of *OTMapOnto*-`global_mnn`, *OTMapOnto*-`global_top20`, and *OTMapOnto*-`refinement`. The results show that the matching candidates derived from the global coupling matrix through either MNN or Top-K achieved high recall but low precision. With the refinement, *OTMapOnto* achieved the best precision and a compatible F1 measure compared to the best tools in the campaign.

Matcher	Threshold	Precision	Recall	F1
AML	0	0.76	0.66	**0.71**
OTMapOnto-`refinement`	0.30	**0.89**	0.59	0.70
GMap	0	0.70	0.67	0.68
LogMap	0	0.78	0.60	0.68
Wiktionary	0	0.78	0.57	0.66
FineTOM	0	0.73	0.58	0.65
TOM	0.86	0.82	0.52	0.64
ATMatcher	0	0.69	0.59	0.64
ALOD2Vec	0.29	0.79	0.54	0.64
edna	0	0.82	0.51	0.63
LogMapLt	0	0.78	0.52	0.62
StringEquiv	0	0.83	0.48	0.61
LSMatch	0	0.83	0.48	0.61
AMD	0	0.81	0.48	0.60
KGMatcher	0	0.83	0.45	0.58
Lily	0.24	0.62	0.52	0.57
OTMapOnto-`global_mnn`	0	0.24	0.73	0.36
OTMapOnto-`global_top20`	0	0.007	**0.96**	0.013

Table 1. Evaluation on OAEI 2021 conference track ordered by f1 in descending order. The shaded value is the best in its category.

Evaluation on MSE Benchmark. The MSE benchmark contains 3 matching cases between 3 materials science and engineering ontologies: MaterialInformation, MatOnto, and EMMO (European Material Modeling ontology). We downloaded the two leading ontology matching systems AML[4] and LogMap[5] for com-

---

[4] https://github.com/AgreementMakerLight/AML-Project
[5] https://github.com/ernestojimenezruiz/logmap-matcher

parison. The results are presented in Table 2. The table shows $OTMapOnto$_refinement achieved the best F1 performance for all 3 test cases.

Test Case	Matcher	Precision	Recall	F1
	$OTMapOnto$-refinement	0.78	0.30	**0.44**
	$OTMapOnto$-global_mnn	0.23	0.39	0.29
Case 1	AML	0.80	0.17	0.29
	LogMap	**1.00**	0.04	0.08
	$OTMapOnto$-global_top20	0.001	**0.78**	0.002
	$OTMapOnto$-refinement	0.31	0.54	**0.39**
Case 2	$OTMapOnto$-global_mnn	0.54	0.26	0.35
	AML	0.82	0.21	0.34
	LogMap	**0.87**	0.20	0.32
	$OTMapOnto$-global_top20	0.01	**0.57**	0.02
	$OTMapOnto$-refinement	**0.98**	0.87	**0.92**
Case 3	AML	0.96	0.87	0.91
	LogMap	0.93	0.84	0.88
	$OTMapOnto$-global_mnn	0.67	0.86	0.75
	$OTMapOnto$-global_top20	0.007	**0.97**	0.014

Table 2. Evaluation on MSE benchmark ordered by f1 in descending order in each case. The shaded value is the best in its category.

8   Discussion

We name our system as $OTMapOnto$ [3]. Our experimental results showed $OTMapOnto$ system achieved promising results. Most significantly, for the MSE benchmarks, $OTMapOnto$ with refinements outperformed the two leading matching systems that have been consistently the top 2 in previous OAEI campaigns in many tracks. $OTMapOnto$ also outperformed almost all of the systems participating in the OAEI 2021 Conference track. This exploratory study provided positive answers to our exploration questions listed in Introduction. First, Wasserstein distance is effective in capturing semantic similarity between ontologies. Second, the coupling matrix returned by the optimal transport solver contains most of the correct matchings but with many spurious ones. Both sets of candidates derived through mutual nearest neighbors (MNN) and Top-K targets (TopK) achieved the best recall results. We also observed this phenomenon in several tasks in the OAEI 2021 Campaign [3], where the results had lower precision and higher recall compared to other systems. Finally, we found that using local Wasserstain distances between the contexts of the source and target concepts of a candidate greatly helps filter out many false positives. The final overall performance metrics are better or compatible to the SOTA results.

In discussing these results, it is important to note that this exploratory study only considered the embeddings of concept labels generated by a pre-trained

model. In future work, we will develop ontology embeddings capturing additional ontology information including structural and logical components. For large scale ontologies, the $n \times m$ matrices associated with optimal transport are unscalable. It is quite evident that we need to break down the ontologies into smaller chunks for computing the optimal transport couplings. In moving forward, we will first partition the embeddings into clusters. Using Wasserstein distances to measure the similarities of pairs of clusters, we will aim to find candidate matchings from the pairs of clusters that are most similar.

9   Related Work

AML [7] and LogMap [17] are two leading classical matching systems based on symbolic structures. Nkisi-Orji et al. in [23] developed an embedding-based supervised machine learning method. Kolyvakis et al. in [18] presented an approach that applied the Stable Marriage algorithm for deriving candidate matchings based on the cosine similarity between retrofitted embedding vectors. Chen et al. in [7] proposed a distantly supervised method that combines embedding-based extensions with classical systems such as AML and LogMap. More recently, He et al. in [16] described a transformer-based ontology matching system, BERTMap. Alvarez-Melis et al. in [1] proposed a method that first encodes hierarchical data in hyperbolic spaces and then applies optimal transport to the hyperbolic embeddings for deriving correspondences. However, the performance for ontology matching is limited when only the hyperbolic embeddings of ontological hierarchies were considered.

10   Conclusion

We explored the effectiveness of the Wasserstein distance metric defined by optimal transport process for ontology matching. Our study showed Wasserstein distance is effective in measuring the similarity between ontologies. The experimental results showed the Wasserstein distance-based approach outperformed almost all the cases in the test data. We plan to test the approach on a wide range of ontology matching applications. In addition, for source and target ontology embedding spaces without 'registration', that is, they do not have well-defined ground distance between them, we will extend to Gromov Wasserstein distance metric [12] which measures how distances between pairs of concepts are matched across ontologies.

References

1. Alvarez-Melis, D., Mroueh, Y., Jaakkola, T.: Unsupervised hierarchy matching with optimal transport over hyperbolic spaces. In: PMLR (2020)
2. An, B., Chen, B., Han, X., Sun, L.: Accurate text-enhanced knowledge graph representation learning. In: NAACL (2018).
3. An, Y., Kalinowski, A., Greenberg, J.: OTMapOnto: Optimal Transport-based Ontology Matching. In: OAEI (2021)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of ACL (2017)
5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS (2013)

6. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: ISWC (2013). pp.
7. Chen, J., Jiménez-Ruiz, E., Horrocks, I., Antonyrajah, D., Hadian, A., Lee, J.: Augmenting ontology alignment by semantic embedding and distant supervision. In: ESWC (2021).
8. Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., Liu, J.: Graph Optimal Transport for Cross-Domain Alignment. arXiv:2006.14744 (Jun 2020)
9. Chen, L., Zhang, Y., Zhang, R., Tao, C., Gan, Z., Zhang, H., Li, B., Shen, D., Chen, C., Carin, L.: Improving sequence-to-sequence learning via optimal transport. In: ICLR (2019).
10. Demeester, T., Rocktäschel, T., Riedel, S.: Lifted rule injection for relation embeddings. In: EMNLP (2016).
11. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: OTM (2013).
12. Gabriel, P., Cuturi, M., Solomon, J.: Gromov-wasserstein averaging of kernel and distance matrices. In: ICML (2016).
13. Galkin, M., Berrendorf, M., Tapley Hoyt, C.: An Open Challenge for Inductive Link Prediction on Knowledge Graphs. arXiv:2203.01520 (2022)
14. Grave, E., Joulin, A., Berthet, Q.: Unsupervised alignment of embeddings with wasserstein procrustes. In: PMLR (2019).
15. Hamilton, W.L.: Graph representation learning. Synthesis Lectures on Articial Intelligence and Machine Learning **14**(3), 1–159 (2020)
16. He, Y., Chen, J., Antonyrajah, D., Horrock, I.: Bertmap: A bert-based ontology alignment system. in: AAAI (2022)
17. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-Based and Scalable Ontology Matching. In: ISWC (2011).
18. Kolyvakis, P., Kalousis, A., Smith, B., Kiritsis, D.: Biomedical ontology alignment: an approach based on representation learning. J Biomed Semant **9**(21) (2018)
19. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI (2015).
20. Lu, F., Cong, P., Huang, X.: Utilizing textual information in knowledge graph embedding: A survey of methods and applications. IEEE Access (2020).
21. Maretic, H.P., Gheche, M.E., Chierchia, G., Frossard, P.: Got: An optimal transport framework for graph comparison. In: NeurIPS (2019).
22. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. CoRR (2013).
23. Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, K.Y., Heaven, R.: Ontology alignment based on word embedding and random forest classification. Machine Learning and Knowledge Discovery in Databases **11051I** (2019)
24. Otero-Cerdeira, L., J., R.M.F., A., G.R.: Ontology matching: A literature review. Expert Systems with Applications **42**(2), 949–971 (2015)
25. Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: NAACL (2015).
26. Veira, N., Keng, B., Padmanabhan, K., Veneris, A.: Unsupervised embedding enhancements of knowledge graphs using textual associations. In: IJCAI (2019).
27. Villani, C.: Optimal Transport: Old and New (Grundlehren der mathematischen Wissenschaften (338)). Springer (2009)
28. Wang, Y., Zhang, H., Shi, G., Liu, Z., Zhou, Q.: A model of text-enhanced knowledge graph representation learning with mutual attention. IEEE Access (2020).
29. Wang, Z., Li, J.Z.: Text-enhanced representation learning for knowledge graph. In: IJCAI (2016)