VeeAlign: Multifaceted Context Representation Using Dual Attention for Ontology Alignment

Vivek Iyer
University of Edinburgh
Edinburgh, UK
viyer@ed.ac.uk

Arvind Agarwal
IBM Research
New Delhi, India
arvagarw@in.ibm.com

Harshit Kumar
IBM Research
New Delhi, India
harshitk@in.ibm.com

Abstract

Ontology Alignment is an important research problem applied to various fields such as data integration, data transfer, data preparation, etc. State-of-the-art (SOTA) Ontology Alignment systems typically use naive domain-dependent approaches with handcrafted rules or domain-specific architectures, making them unscalable and inefficient. In this work, we propose VeeAlign, a Deep Learning based model that uses a novel dual-attention mechanism to compute the contextualized representation of a concept which, in turn, is used to discover alignments. By doing this, not only is our approach able to exploit both syntactic and semantic information encoded in ontologies, it is also, by design, flexible and scalable to different domains with minimal effort. We evaluate our model on four different datasets from different domains and languages, and establish its superiority through these results as well as detailed ablation studies. The code and datasets used are available at https://github.com/Remorax/VeeAlign.

1 Introduction

Ontology alignment is the task of establishing correspondences between semantically related elements (i.e. classes and properties) from different ontologies. It is useful for many applications, particularly for data integration and data migration. The problem has been extensively studied in the past decade, and the solutions have ranged from simple rule based systems (Faria et al., 2013; Jiang et al., 2016) to the ones incorporating external knowledge (Hertling and Paulheim, 2012; Algergawy et al., 2011), as well as the most recent ones that use sophisticated deep learning based systems (Kolyvakis et al., 2018; Wang et al., 2018; Jiménez-Ruiz et al., 2020; Xue et al., 2021). A common limitation of these systems is their inability to generalize to new data - rule based systems are based on handcrafted rules which may not cover all data scenarios in all datasets. On the other hand, the Deep Learning (DL) based systems (Kolyvakis et al., 2018; Wang et al., 2018) proposed so far, often have strong dependencies on domain-specific external knowledge bases such as lexicons and thesauri. Moreover, they also underperform as compared to their rule-based counterparts. One of the primary reasons for this and in fact, also the dependency of DL architectures on external background knowledge is the lack of sufficient, usable training data. Ground truth alignments are typically very scarce in number, especially for smaller ontologies, making supervised training difficult. Moreover, the small number of "positive" alignment pairs as compared to the "negative" ones (i.e entity pairs that contain an alignment versus those that do not) leads to a class imbalance and further adds to the difficulties of supervised learning based solely on the reference alignments. For example, the conference dataset (Zamazal and Svátek, 2017), used for experimentation in this paper, has 305 positive and 122588 negative alignments. As a result of both sparsity and class imbalance, even moderately complex DL architectures (that contain only a few parameters) overfit and therefore, perform poorly. Given these challenges and the weaknesses of the previous approaches, our goal in this paper is two fold: a) to build a generic, domain-independent model that leverages the intrinsic semantic and structural information encoded in ontologies with no requirement of external, domain-specific knowledge and b) a model that uses a parametrically-light architecture to strike the right balance between the model expressivity (uses training data well) and model complexity (does not overfit).

Despite significant research, ontology alignment still remains a challenging task. Figure 1 provides an illustration highlighting this challenge. The task is to determine alignment between the concept *Attendee* in Ontology-1 and the concept *Listener* in Ontology-2. Current approaches that com-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10780–10792
November 7–11, 2021. ©2021 Association for Computational Linguistics

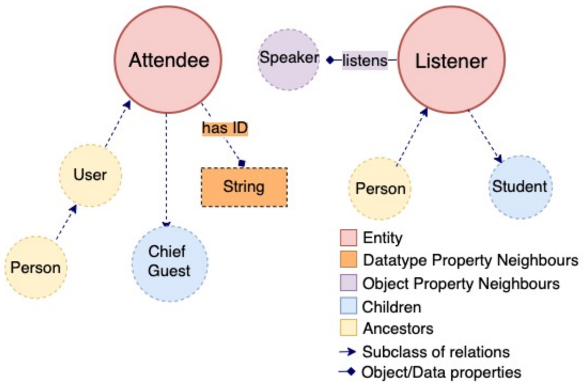

Figure 1: An Example illustrating concept alignment and dependency on the surrounding context

pute concept similarity or context similarity at the label level will fail to capture the alignment between these two concepts, since these concepts do not have high lexical or even semantic similarity. While there is a common concept (i.e. *Person*) in the neighbouring context of both concepts, there are several other concepts in their respective contexts that are not similar. This example shows that not only is it important to consider neighbouring concepts, it is also important to model them in such a way that while computing similarity, the relevant neighbouring concepts have higher weights than the irrelevant ones. In this particular example, ancestor nodes should be given higher weights than children and concepts connected by object and datatype properties respectively.

Driven by this intuition and the need to address the limitations of the current approaches, this paper presents a novel ontology alignment system, referred to as VeeAlign[1], that exploits both syntactic (context surrounding the concepts) as well as semantic (label-based similarity) aspects of an ontology to compute alignment. We present a novel way of modelling context, where the context is split into multiple facets based on the type of contextual concepts it contains[2]. More specifically, we split the context into four facets, where each facet contains contextual concepts that have a different relationship with the central concept. This includes facets containing ancestors, children, contextual concepts connected by object properties and those connected by datatype properties respectively. Such a multi-faceted context, however, poses a new challenge because not all of these facets contribute equally in the relatedness of the concepts. Moreover, each of these facets consist of paths which, in turn, are composed of nodes. Therefore, the challenge here lies in developing a mechanism that weighs the relevant parts of the context (be it facets, or paths, or nodes, in increasing order of granularity), more than the less relevant ones. In order to deal with this challenge, we propose a novel dual attention mechanism comprising of a) path-level attention and b) node-level attention. The path-level attention combines all the path representations in a facet based on their importance (to the alignment of the central concept) to compute an aggregated path representation. Similarly, the node-level attention combines all node representations in the aggregated path based on their importance to return an aggregated node representation. We apply these attention mechanisms sequentially on each facet to obtain its representation. The final context representation is obtained using a weighted sum of these facet representations, where the weights are proportional to their importance. We also note that the term “dual attention" is rather overloaded in prior literature, so we distinguish its usage in this work from that of prior works (Fu et al., 2019; Nam et al., 2017; Yan et al., 2019; Liu et al., 2019). In these works, dual attention has either referred to the parallel application of separate attention mechanisms or simultaneous application on two or more different features. Our usage of dual attention lies in the *sequential* application of attention on the same features, first at the path level and then at the node level. The main contributions of this paper are, therefore, as follows:

- We model the task of ontology alignment to determine similarity between two concepts, with a major focus on context. We introduce the notion of multi-faceted context, and model it using a novel dual attention mechanism.
- We show through an ablation study the effect of dual attention over single attention and no attention, and the effect of different facets on model performance.
- To demonstrate the applicability of our approach on diverse data sources in terms of language, domain, and size, we evaluate the proposed model on four different datasets: Conference, Lebensmittel, Freizeit, and Web Directory. We show that our approach of context modelling outperforms SOTA baselines on all datasets, sometimes by a significant margin, and in particular, significantly increases recall of the positive alignments.

[1] https://github.com/Remorax/VeeAlign

[2] A facet is a set of concepts belonging to same category.

2 Related Work

Related work mainly spans two broad areas: ontology and Knowledge Graph (KG) alignment.

Ontology Alignment There is a large body of work on ontology alignment (Euzenat et al., 2007; Otero-Cerdeira et al., 2015; Niepert et al., 2010; Schumann and Lécué, 2015), primarily driven by the Ontology Alignment Evaluation Initiative (OAEI). OAEI has been organising ontology alignment challenges since 2004 where multiple datasets belonging to different domains are released along with a public evaluation platform to evaluate different systems. Among all systems submitted to the challenge, AgreementMakerLight (AML) (Faria et al., 2016) and LogMap2 (Jiménez-Ruiz et al., 2020) have consistently outperformed other systems for the past several years. Despite their high performance on OAEI datasets, these systems have two identifiable weaknesses: a) For comparing concept labels, string similarity measures are used which do not address semantic similarity, and b) While these systems use neighbour alignment as a guiding factor for calculating alignments, they do not attempt to compute context similarity using a rigorous neighbourhood representation. In addition, while they have been engineered over the years to give the best performance on OAEI datasets, their performance is less impressive on non-OAEI datasets (refer Section 4), indicating poor scalability. Wiktionary (Portisch et al., 2020) is another top performing ontology alignment system, especially in the multilingual space. However, it relies heavily on the Wiktionary knowledge base which again presents scalability issues.

The few systems proposed outside OAEI, particularly Deep Learning based systems, have strong dependencies on background knowledge sources, thus limiting their use. A good example of this is OntoEmma (Wang et al., 2018), a neural network based ontology alignment system for the Biomedical domain. It enriches ontological entities with aliases from the ontology, definitions from Wikipedia, and usage contexts from domain-specific medical papers; and uses this additional information for ontology alignment. Similarly, DeepAlign (Kolyvakis et al., 2018) also requires synonyms and antonyms extracted from external sources such as WordNet & PPDB in order to refine word vectors using synonymy and antonymy constraints, which are then used for alignment. Finally, the recently proposed SNN-OM technique (Xue et al., 2021) uses, among others, WordNet-dependent similarity features for aligning sensor ontologies.

A recent work attempting to combine the traditional and modern DL-based approaches proposed a Machine Learning-based extension to traditional ontology alignment systems, using distant supervision, ontology embeddings and Siamese Neural Networks (Chen et al., 2021). While this was useful in incorporating richer semantics and outperforming the traditional systems, it was still reported to underperform as compared to VeeAlign.

KG Alignment While Deep Learning-based research in ontology alignment has been rather limited, there has been quite a lot of work exploring entity alignment in Knowledge Graphs, particularly in the last 2 years (Zhang et al., 2021). Sun et al. (Sun et al., 2017) proposed the DBP15K dataset that provided cross-lingual entity alignments between Japanese-English, French-English and Chinese-English versions of DBPedia respectively. This work has led to the proposal of various entity alignment systems (Sun et al., 2017, 2020; Mao et al., 2020; Liu et al., 2021a; Nguyen et al., 2020a; Liu et al., 2021b; Lu et al., 2021; Qi et al., 2021) that, given a pair of Knowledge Graphs (KGs), seek to discover an injective mapping between the entities of the corresponding KGs. The common approach used by these systems involves ranking the most similar entities in KG-2 for each entity in KG-1. These systems have achieved remarkable success in entity alignment in KGs, with EMGCN (Nguyen et al., 2020b) emerging as the best-performing system on the leaderboards for DBP15K Zh-en (PapersWithCode, 2021c), DBP15K Fr-en (PapersWithCode, 2021a) as well as DBP15K Ja-en (PapersWithCode, 2021b). EMGCN uses an unsupervised entity alignment framework that exploits various aspects of KG-specific data and combines them via a late-fusion mechanism. Despite the success of these systems in KG entity alignment, particularly in DBPedia, applying these systems for ontology alignment presents certain significant problems. Firstly, they assume that for each entity in the source KG, there would be a matching entity in the target KG. Some systems, like DGMC (Liu et al., 2021a), additionally also employ the principle of ‘neighborhood consensus’ to train their systems: neighbours of aligned entities must also contain corresponding alignments in their neighbourhoods. PRASE (Qi et al., 2021), which combines probabilistic reasoning and semantic embedding for entity and relation alignment, uses a similar technique for obtaining seed alignments using its PARIS-based reasoning system. These assumptions of neighbourhood similarity are intuitive and effective while aligning linguistic variants sourced from the same KG, such as En-Fr DBPedia. However, these assumptions are less valid and effective while aligning differently-sourced ontologies possessing dissimilar structures. More importantly, the KG entity alignment systems require large amounts of training data. The large number of interlingual alignments present in DBP15K dataset can satisfy this constraint. In addition, DBPedia provides additional training data due to its richness and denseness. For instance, EMGCN trains on the diverse, numerous attributes of DBPedia concepts while EVA (Liu et al., 2021b) trains on the stored images for computing visual embeddings used during alignment. These methods, while suitable for large KGs, may not be very suitable for ontology alignment datasets, where the alignments are typically sparse and much fewer in number. In addition, small-medium size ontologies often do not possess significant concept-level information, resulting in substantially lesser training data for complex DL architectures.

In this work, we attempt to resolve drawbacks present in both rule-based Ontology Alignment systems that require extensive manual effort, as well as DL-based complex KG entity alignment systems requiring extensive training data. VeeAlign uses a light and robust DL architecture that both increases expressivity over Ontology Alignment systems and also minimises training data required.

3 Approach

In this section, we describe our system VeeAlign including its underlying dual attention mechanism.

3.1 Preliminaries

Let O^s and O^t be the source and target ontologies with the corresponding concepts $\{c_1^s, c_2^s, \dots c_M^s\}$ and $\{c_1^t, c_2^t, \dots c_N^t\}$, respectively. Ontology alignment in its most general form involves finding different kinds of relationships between concept pairs, including complex relationships such as transformation (Thiéblin et al., 2020) or inference (Zhou, 2018). The focus of this work is to discover the equivalence relationship between concepts, primarily because they are of the most interest to the community. Terminologically, we refer to concepts being compared for alignment as central concepts, and the concepts surrounding a central concept as contextual concepts (or context as a whole). Our approach for finding semantically equivalent concepts involves computing the representations of the central concept and its context, and then combining them for discovering alignments.

3.2 Concept Representation

We illustrate VeeAlign's architecture in Figure 2. Since VeeAlign is a supervised model, it requires training data in the form of both positive (aligned) and negative (non-aligned) concept pairs. So, for a given source and target ontology, we have a training set $\mathcal{T}$ consisting of concept pairs (e.g. (c_i^s, c_j^t)) along with their labels ($L(c_i^s, c_j^t)$), where label is 1 when c_i^s and c_j^t are semantically equivalent and 0 otherwise. VeeAlign computes the concept representation using pre-trained language model (e.g. Universal Sentence Encoder). The key difference however lies in its method of capturing the multifaceted context and computing a contextualized concept representation which is explained below.

3.3 Context Representation

Our hypothesis is that the context plays a critical role in alignment, therefore, it is important to model the context in a principled manner. The context of a central concept consists of all the surrounding concepts referred as contextual concepts. For a concept c_i, let u_i be its d-dimensional distributed representation obtained using the Universal Sentence Encoder (Cer et al., 2018). Each contextual concept has a role and influences the alignment of the central concept, therefore we categorize contextual concepts into 4 facets: ancestral concepts, child concepts, concepts connected through an object property and concepts connected through a datatype property.

Sifting through several ontologies and their reference alignments, we observed that two concepts align not only on the basis of their one-hop neighbours but also on the similarity of their “ancestral concepts". In other words, while comparing two concepts, we consider not only their immediate parents but also the long-range ancestral concepts that lie on the path (also referred to as lineage path) from the central concept all the way to the root concept. We thus enumerate all lineage paths from the central concept to the root and use them for

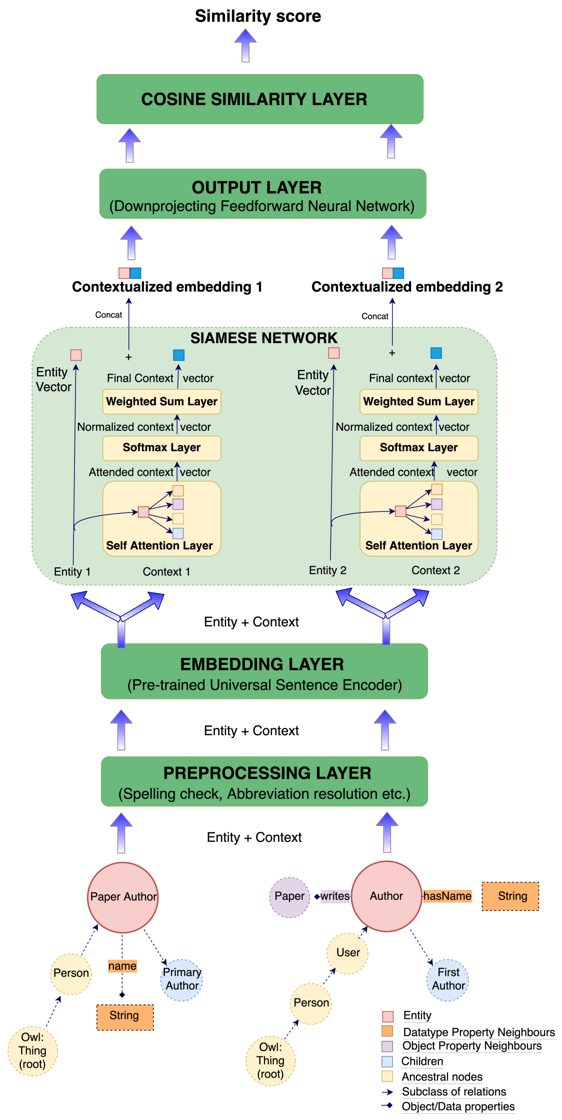

Figure 2: VeeAlign Architecture

context representation. Concepts that link to the central concept using a direct SUBCLASS-OF relationship are known as child concepts. In order to follow consistent terminology, we also represent these as a path, however, each path in this case contains only one concept. The contextual concepts that link to the central concept through datatype and object properties are represented in the same manner as child concepts, i.e. we only consider one-hop neighbours linked to the central concept through either of the two properties. Also, for the sake of uniformity with subclass relations, we only consider the neighbours linked through these properties while computing representations and not the properties themselves.

3.4 Dual Attention

Attention (Bahdanau et al., 2015) in deep learning can be broadly interpreted as a vector of weights denoting relative importance. Here, attention computes weights for contextual concepts that determine its directly proportional influence on the alignment of the central concept. To compute the weights, we use a dual attention mechanism that consists of two steps: i) Path-level attention and ii) Node-level attention. The goal of the first step is to assign higher weights to the most influential path(s), and then using weighted average to compute a unified path representation, whereas in second, the goal is to assign higher weights to the nodes in the computed unified path that are most influential in the alignment of the central concept.

Path-level Attention As mentioned, the path level attention aims to find the most important paths in each facet. The weight of each path is computed as the sum of its node weights, and this is detailed as follows. Among the four different facets, let us first consider the facet containing ancestral concepts and specifically, the lineage paths, which are essentially paths of ancestral concepts from the central concept to the root. For the central concept c_i, let it contain n lineage paths $P_{i1}, P_{i2}, \ldots P_{ij}, \ldots, P_{in}$. Let $c_{ij1}, c_{ij2}, \ldots c_{ijk}, \ldots, c_{ijt}$ be the t concepts in j^{th} path of length t. Here i, j, and k are indices for the central node, its neighboring paths, and concepts in those paths respectively. Now, let the maximum length among all n neighbouring paths be l. Here, $t \leq l$, so all the paths are padded with null concepts (represented using zero vectors) appropriately, so as to ensure each path is of length l. The attention weights for each concept in each path are then calculated as a dot product of the central concept and the path concept, i.e:

$$w_{ijk} = u_i^T u_{ijk} \tag{1}$$

These are then summed to obtain the overall weight of a path i.e.,

$$w_{ij} = \sum_k w_{ijk}. \tag{2}$$

Once the relative importance of each path is computed, the next step involves obtaining a unified path representation as a weighted average of all the paths. Let $w_{i1}, w_{i2}, w_{ij} \ldots$ be the relative importance of all the lineage paths obtained from (2), then

$$R_{ik} = \sum_{j} w_{ij} u_{ijk} \tag{3}$$

Here R_{ik} is the d-dimensional representation of the k_{th} node on the unified path representation of i_{th} central concept. The unified path representation for the i_{th} central concept is, therefore, the stacked $l * d$-dimensional matrix R_i where:

$$R_i = [R_{i1}, R_{i2}, \ldots, R_{ik}, \ldots, R_{il}] \tag{4}$$

Node-level Attention Each node (or contextual concept) in the path contributes towards the central concept's alignment proportional to its importance, which is determined by the node-level attention. Thus in this second step of attention, node weight is determined as:

$$w_{ik} = \text{Softmax}(u_i^T R_{ik}) \tag{5}$$

After computing node-level attention, these node weights are then used to compute the final representation of a facet as follows:

$$F_i = \sum_{k} \theta_k \, w_{ik} \, R_{ik} \tag{6}$$

where F_i is the d-dimensional final representation vector of a facet. θ_k is a scalar introduced to provide importance to each contextual concept based on its distance/positional index (k in this case) from the central concept. This is driven by the intuition that immediate ancestors play a more important role in alignment than the distant ones.

Thus in this way, we follow a sequential dual attention approach to compute the representation of each facet in the context. The computations for learning the representations of ancestral concepts and the other three facets of concepts is same, except that, for the other three facets the path length is one. Again, appropriate padding with null concepts is done to ensure that all paths, be it lineage paths or the paths connecting one-hop neighbours, are of the same length.

Final Context Representation Having computed the representation for each facet, we now proceed to calculate final context representation. Let F_{ia}, F_{ic}, F_{io}, and F_{id} be the facet representations obtained using equation 6 for ancestral concepts, child concepts, concepts connected through object properties and those connected through the datatype properties respectively. The final context representation is obtained as a weighted sum of the facet representations as follows:

$$\begin{aligned} v_i &= fw_a F_{ia} + fw_o F_{io} + fw_c F_{ic} + fw_d F_{id} \\ s.t. \quad & fw_a + fw_o + fw_c + fw_d = 1 \end{aligned} \tag{7}$$

Where fw_a, fw_o, fw_c, fw_d are the weights for the corresponding facet representations respectively.

Training Layer This context representation v_i is concatenated with the central concept representation u_i, and then passed to a feedforward Neural Network for dimensional reduction as

$$f(c_i) = W * [u_i, v_i]. \tag{8}$$

Here $f(c_i)$ is the final *contextualized* representation of the central concept c_i.

For the property alignment, we do not use context and simply use semantic representations of the property name. For a given property p_i, $g(p_i)$ is the d-dimensional representation provided by the embedding layer. A candidate alignment pair consists of elements of similar type (concepts with concepts and properties with properties) from both source and target ontologies. The aforementioned computations are performed for both source and target elements by passing them both through a Siamese Network (Bromley et al., 1994) (refer Figure 2) and then computing the confidence score of the alignment by taking a cosine similarity between the two contextualized representations, i.e.

$$\begin{aligned} H(c_i^s, c_j^t) &= \cos(f(c_i^s), f(c_j^t)) \\ H(p_i^s, p_j^t) &= \cos(g(p_i^s), g(p_j^t)) \end{aligned} \tag{9}$$

Denoting concept and property pairs as element pairs (e_i^s, e_j^t), an element pair is considered a positive alignment when the similarity score is more than a threshold parameter Θ, i.e. $\hat{L}(e_i^s, e_j^t) = 1$ when $H(e_i^s, e_j^t) > \Theta$ and 0 otherwise. For the training, we use mean squared error,

$$\mathcal{L} = \frac{1}{T} \sum_{(e_i^s, e_j^t) \in \mathcal{T}} \Big(H(e_i^s, e_j^t) - L(e_i^s, e_j^t) \Big)^2$$

where $H(e_i^s, e_j^t)$ is obtained using equation (9), and T is total number of training examples. $L(e_i^s, e_j^t)$ denotes the ground truth label which is 1 if $e_i^s \equiv e_j^t$ and 0 otherwise. The architecture has a relatively small parameter footprint with learnable parameters being used only in equations (7) and (8) which is 307204. This is by design and is driven to balance the trade-off between model complexity and expressivity.

4 Experiments

This section provides experiment details, i.e. the datasets used, baseline models, experimental setup, results, and their analysis including ablation study.

4.1 Datasets

We evaluate the performance of our model on four benchmark datasets used in several prior studies for the ontology alignment task ((Euzenat et al., 2011; Peukert et al., 2010)). Table 1 shows the number of concepts in each ontology along with the total number of ground truth positive alignments. The Conference (Zamazal and Svátek, 2017) dataset is an English language dataset from the conference organization domain. The other three datasets are in German Language. Lebensmittel (Peukert et al., 2010) is from Food domain, whereas Freizeit (Peukert et al., 2010) and Web directory (Massmann and Rahm, 2008) are from online shopping domain.

Dataset	Ontology	#Concepts	#Ground Truth Alignments
Conference	cmt	29	305
	conference	59	
	confOf	38	
	edas	103	
	ekaw	73	
	iasted	140	
	sigkdd	49	
Lebensmittel	Google	58	32
	web	52	
Freizeit	dmoz	70	67
	Google	66	
Web directory	dmoz	745	2051
	Google	727	
	web	417	
	Yahoo	1132	

Table 1: Datasets used in the experiments

4.2 Hyperparameters

We optimised hyperparameters through grid-search. The word vectors were initialized with 512-dimension Universal Sentence Encoder (USE) (Cer et al., 2018) for the conference dataset and its multilingual variant(Yang et al., 2020) for the three German-language datasets. The model was converged using MSE loss and Adam optimizer with a learning rate of 0.001 and a batch size of 32. Model training was stopped after a maximum of 50 epochs. For obtaining unified path representation using equation 2, we experiment with weighted sum and max pooling, and report the best results. Finally, the dimension of the final output layer was set to 300. For reproducibility, all randomizations were seeded with 0, and the system was run only once. More details on the experimental setup including computing infrastructure is provided in Appendix A.

4.3 Data Preprocessing and Evaluation Methodology

VeeAlign needs both positive and negative alignment pairs, and since datasets only come with positive pairs, all other possible non-positive pairs are considered as negative alignment pairs. To prevent class imbalance due to the high number of negative alignment pairs compared to positive ones, the positive pairs are oversampled to match the negative ones. The entire data (positive and negative pairs) is split into training, validation and test sets in 70:20:10 ratio using the K-fold “sliding window" method. To ensure sufficient training data despite the varied size of the datasets, we use $K = 7$ in conference and $K = 5$ in the other 3 datasets. Further details are provided in Appendix A. The validation set is used for optimizing hyperparameters including the classification threshold. Precision, recall and F1-score of the positive class are used as evaluation metrics.

4.4 Results and Discussion

Tables 2, 3, 4 and 5 show the evaluation results on Conference, Lebensmittel, Freizeit and Web directory datasets, respectively. We evaluate the performance of VeeAlign by comparing it with the following top performing baselines from different areas: AML(Faria et al., 2016), LogMap2 (Jiménez-Ruiz et al., 2020), Wiktionary(Portisch et al., 2020), DeepAlign (Kolyvakis et al., 2018) and the KG alignment system EMGCN (Nguyen et al., 2020b). These baselines were selected based on their performance (both, top performers in OAEI and outside OAEI), open-source availability and in an effort to cover both DL and non-DL based methods. Particularly, AML and LogMap have consistently been top performers for several years in OAEI, and continue to retain their standing. Recently, the Deep Learning-based system DeepAlign beat AML on the conference dataset, while Wiktionary has emerged as a close competitor to AML and LogMap, particularly in multilingual tasks. EMGCN is a leaderboard topper among the KG entity alignment systems. For more details on these baselines, refer to Section 2.

While we attempted to run all baselines on all datasets using their original setting, some baselines failed to produce results in some datasets. In particular, AML timed out on the Web directory dataset, possibly due to its large size and the unoptimized nature of string computations on large non-OAEI datasets. It finishes in time for the relatively much smaller Lebensmittel and Freizeit datasets, but only outputs instance alignments and fails to discover any concept alignments. Additionally, since DeepAlign employs external English-language lexicons for refining word vectors, it could not be run on German-language datasets. Lastly, EMGCN could not be run on the conference dataset. This is because the current implementation of EMGCN, that involves message-passing GCNs, cannot support ontologies with disconnected concepts, such as those present in conference dataset.

System	P	R	F
AML	0.802	0.651	0.700
LogMap2	0.821	0.654	0.701
Wiktionary	0.685	0.608	0.629
DeepAlign	0.631	0.586	0.567
VeeAlign	0.774	0.741	**0.748**

Table 2: Results on the Conference dataset

From the results in Tables 2, 3, 4 and 5, VeeAlign outperforms the baselines on all four datasets, often by a huge margin. On comparison against the second best performing baseline in Conference, Lebensmittel, Freizeit, and Web Directory, there are improvements of 6.7%, 52.8%, 0.11%, and 14.4%, respectively. Note that the results in Table 3 have a slightly wider range of values which is primarily due to the relatively small number of alignment pairs in this dataset.

System	P	R	F
AML	0.000	0.000	na
EMGCN	0.011	0.273	0.020
LogMap2	1.000	0.300	0.437
Wiktionary	1.000	0.300	0.437
VeeAlign	0.889	0.540	**0.668**

Table 3: Results on the Lebensmittel dataset

System	P	R	F
AML	0.000	0.000	na
EMGCN	0.013	0.196	0.023
LogMap2	0.925	0.747	0.821
Wiktionary	0.803	0.969	0.878
VeeAlign	0.814	0.970	**0.879**

Table 4: Results on the Freizeit dataset

System	P	R	F
AML	Timed out	Timed out	Timed out
EMGCN	0.001	0.030	0.002
LogMap2	0.778	0.542	0.638
Wiktionary	0.503	0.665	0.573
VeeAlign	0.746	0.741	**0.730**

Table 5: Results on the Web Directory dataset

These results lead us to the following four conclusions: i) The superior performance of VeeAlign on all 4 datasets indicates the efficacy and robustness of our algorithm, irrespective of size, domain or language of the dataset. ii) Though AML has long been the SOTA in ontology alignment, its inapplicability to non-OAEI baselines, namely in terms of execution time and lack of detection of concept alignments, indicates lack of scalability as a potential (and major) weakness. DeepAlign in turn, which was the previous SOTA in DL alignment systems, requires synonymy and antonymy information provided as background knowledge, which too can lead to scalability issues due to unavailability of the same. iii) EMGCN performs poorly on all 3 German datasets. This is likely caused by a variety of reasons. Firstly, the ontologies in the German datasets are sparsely connected and possess low node degree (1.8) as compared to the average DBPedia degree[3] (18.85). Secondly, DBPedia is much richer in terms of attributes: the number of attributes in the provided DBPedia KGs (4̃000) and the number of attribute triples (200,000-300,000) far outweigh the number of attributes (2-3) and attribute triples (150-200) in these ontologies respectively. As the provision of relations and attributes is crucial to structural and representational training of EMGCN respectively, the model is possibly highly undertrained. iv) It is also interesting to note that in most cases, Deep Learning (DL) based systems enjoy a higher degree of recall, while rule-based matchers enjoy higher precision. This is intuitive given rule-based matchers deploy rules that always work for certain scenarios, generating high precision, but DL isn't bound by such rules and is thus able to cover relatively more scenarios.

[3] http://konect.cc/networks/dbpedia-link/

	P	R	F
No Context	0.775	0.608	0.670
Context + single attention	0.678	0.728	0.697
Context + dual attention	0.774	0.741	0.748

Table 6: Effect of context in single and dual attention

4.5 Ablation Study

We now perform ablation studies to evaluate the effect of context, dual attention and types of facets on alignment respectively.

Effect of Context and Attention We first analyze the effect of context, and attention on model performance, namely in: i) the absence of context; ii) the presence of context but only single attention (i.e no path level information, only node-level information from neighbours) and iii) the presence of context with dual attention i.e., using both path-level and node-level information. Results in Table 6 indicate that adding context helps in improving performance. Furthermore, the superior performance of dual attention over single attention also proves the utility of dual attention in obtaining better contextual representations.

Facets	P	R	F
Ancestor Concepts	0.747	0.707	0.719
Child Concepts	0.634	0.750	0.678
Object Properties	0.647	0.740	0.681
Data Properties	0.640	0.750	0.681
All combined	0.774	0.741	0.748

Table 7: Effect of Facets on VeeAlign

Effect of Facets We now proceed to discover the importance of each facet in deciding alignment of the central concept. Accordingly, we evaluate the performance using information from only ancestor concepts, child concepts, datatype property neighbours and object property neighbours and compare this against the VeeAlign model that uses all four of them together (Table 7). These results indicate that ancestor concepts are the most useful facet whereas the child concepts are least useful. However the best alignment results are obtained when we combine all four facets.

4.6 Time Complexity

Our algorithm has a run time complexity of $O(mn)$ where m and n are the size of source and target ontologies respectively. Empirically, for the largest ontology pair (yahoo-dmoz) from web-directory dataset with $m = 1132$ and $n = 745$, it took 93 seconds to run. While the algorithm is able to run in reasonable time for ontologies of moderate size, it becomes challenging for large ontologies given its quadratic complexity. One of the future works is to reduce this quadratic complexity by an intelligent selection of target candidates for alignment, thus reducing the search space from n to a constant k.

5 Conclusion

This paper presents a general purpose ontology alignment system that does not require any external or background knowledge. It is based on a light and robust DL architecture that utilises a novel multi-faceted context representation approach for exploiting the structural aspects of an ontology. A novel dual attention mechanism is proposed for focusing on the parts of the context that are most crucial for alignment. Our experiments on 4 different datasets from 2 different languages and 3 different domains show that the proposed method outperforms the SOTA methods by a significant margin. Ablation study examines the effect of context splitting and dual attention, and validate them as the right factors behind the performance improvement.

Acknowledgements

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 825299 (GoURMET).

References

Alsayed Algergawy, Sabine Massmann, and Erhard Rahm. 2011. A clustering-based approach for large-scale ontology matching. In *East European Conference on Advances in Databases and Information Systems*, pages 415–428. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, Denvar Antonyrajah, Ali Hadian, and Jaehun Lee. 2021. Augmenting ontology alignment by semantic embedding and distant supervision. In *European Semantic Web Conference*, pages 392–408. Springer.

Jérôme Euzenat, Christian Meilicke, Heiner Stuckenschmidt, Pavel Shvaiko, and Cássia Trojahn. 2011. Ontology alignment evaluation initiative: six years of experience. In *Journal on data semantics XV*, pages 158–192. Springer.

Jérôme Euzenat, Pavel Shvaiko, et al. 2007. *Ontology matching*, volume 18. Springer.

Daniel Faria, Catia Pesquita, Booma S Balasubramani, Catarina Martins, Joao Cardoso, Hugo Curado, Francisco M Couto, and Isabel F Cruz. 2016. Oaei 2016 results of aml. In *11th international workshop on ontology matching co-located with the 15th international semantic web conference, CEUR workshop proceedings*, volume 1766.

Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F Cruz, and Francisco M Couto. 2013. The agreementmakerlight ontology matching system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 527–541. Springer.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154.

Sven Hertling and Heiko Paulheim. 2012. Wikimatch: using wikipedia for ontology matching. *Ontology Matching*, 946.

Shangpu Jiang, Daniel Lowd, Sabin Kafle, and Dejing Dou. 2016. Ontology matching with knowledge rules. In *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXVIII*, pages 75–95. Springer.

Ernesto Jiménez-Ruiz, Asan Agibetov, Jiaoyan Chen, Matthias Samwald, and Valerie Cross. 2020. Dividing the ontology alignment task with semantic embeddings and logic-based modules. In *ECAI*, pages 784–791.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. 2018. 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, Volume 1 (Long Papers)*, pages 787–798.

Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 344–352.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021a. Visual pivoting for (unsupervised) entity alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4257–4266.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021b. Visual pivoting for (unsupervised) entity alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Guoming Lu, Lizong Zhang, Minjie Jin, Pancheng Li, and Xi Huang. 2021. Entity alignment via knowledge embedding and type matching constraints for knowledge graph inference. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11.

Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational reflection entity alignment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1095–1104.

Sabine Massmann and Erhard Rahm. 2008. Evaluating instance-based matching of web directories. In *WebDB*.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307.

Tam Thanh Nguyen, Thanh Trung Huynh, Hongzhi Yin, Vinh Van Tong, Darnbi Sakong, Bolong Zheng, and Quoc Viet Hung Nguyen. 2020a. Entity alignment for knowledge graphs with multi-order convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Tam Thanh Nguyen, Thanh Trung Huynh, Hongzhi Yin, Vinh Van Tong, Darnbi Sakong, Bolong Zheng, and Quoc Viet Hung Nguyen. 2020b. Entity alignment for knowledge graphs with multi-order convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*.

Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. 2010. A probabilistic-logical framework for ontology matching. In *AAAI*, pages 1413–1418. Citeseer.

Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. 2015. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971.

PapersWithCode. 2021a. Entity Alignment on DBP15k fr-en. https://paperswithcode.com/sota/entity-alignment-on-dbp15k-fr-en. Online; accessed 16-May-2021.

PapersWithCode. 2021b. Entity Alignment on DBP15k ja-en. https://paperswithcode.com/sota/entity-alignment-on-dbp15k-ja-en. Online; accessed 16-May-2021.

PapersWithCode. 2021c. Entity Alignment on DBP15k zh-en. https://paperswithcode.com/sota/entity-alignment-on-dbp15k-zh-en. Online; accessed 16-May-2021.

Eric Peukert, Sabine Massmann, and Kathleen Koenig. 2010. Comparing similarity combination methods for schema matching. *INFORMATIK 2010. Service Science–Neue Perspektiven für die Informatik. Band 1*.

Jan Portisch, Michael Hladik, and Heiko Paulheim. 2020. Wiktionary matcher. In *CEUR Workshop Proceedings*, volume 2536, pages 181–188. RWTH.

Zhiyuan Qi, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Yuejia Xiang, Ningyu Zhang, and Yefeng Zheng. 2021. Unsupervised knowledge graph alignment by probabilistic reasoning and semantic embedding. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2019–2025. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Anika Schumann and Freddy Lécué. 2015. Minimizing user involvement for accurate ontology matching problems. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1576–1582.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644. Springer.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 222–229.

Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. 2020. Survey on complex ontology matching. *Semantic Web*, 11(4):689–727.

Lucy Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, and Waleed Ammar. 2018. Ontology alignment in the biomedical domain using entity definitions and context. In *Proceedings of the BioNLP 2018 workshop*, pages 47–55, Melbourne, Australia. Association for Computational Linguistics.

Xingsi Xue, Chao Jiang, Jie Zhang, Hai Zhu, and Chaofan Yang. 2021. Matching sensor ontologies through siamese neural networks without using reference alignment. *PeerJ Computer Science*, 7:e602.

Shipeng Yan, Songyang Zhang, Xuming He, et al. 2019. A dual attention network with semantic embedding for few-shot learning. In *AAAI*, pages 9079–9086.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Ondřej Zamazal and Vojtěch Svátek. 2017. The ten-year ontofarm and its fertilization within the onto-sphere. *Journal of Web Semantics*, 43:46–53.

Rui Zhang, Bayu Distiawan Trisedy, Miao Li, Yong Jiang, and Jianzhong Qi. 2021. A comprehensive survey on knowledge graph entity alignment via representation learning. *arXiv preprint arXiv:2103.15059*.

Lu Zhou. 2018. A journey from simple to complex alignment on real-world ontologies. In *DC@ ISWC*, pages 93–101.

A Appendix

A.1 Dataset split

For the conference dataset, we do 7-fold cross validation, so out of 21 ontology pairs, 6 folds (18 pairs) are used for training, 1 fold for validation and testing (2 pairs from 1 fold for validation and 1 pair for testing). For the Lebensmittel, Freizeit and Web directory datasets, we perform 5-fold cross validation, in which 70% of the concept pair alignments are used for training, 10% for validation and 20% for testing. Since the conference dataset consists of 21 pairs of small ontologies, we split them at the ontology-pair level. Whereas, since Lebensmittel, Freizeit and Web Directory datasets consist of 1, 1 and 6 pairs of ontology alignments respectively, we split them at the concept-pair level in order to obtain reasonable amounts of training data for facilitating the training process.

A.2 Parameters chosen

There are several parameters that are input to our algorithm. Some of those parameters come from the data characteristics, some are design choices (and therefore fixed), and some are fixed based on prior experience. However there are few parameters which are variable, and we conduct experiments over different values of these parameters and report the best results (in terms of the best F1-score) after performing K-fold sliding window evaluation. These parameters are listed in Table 8. Among the variable parameters, the first parameter, BAG_OF_NEIGHBOURS is what determines how we represent the concept's one-hop neighbours. We have two options for this, either consider them as a path of length one, or bag them together. In the case of the former, we further experiment over the optimal number of paths by using a wide range of values from 1 to the maximum number of paths possible, which is 38 for conference, 16 for Lebensmittel, 12 for Freizeit and 49 for the web directory datasets respectively. Then, to determine optimal length of path, we experiment from 1 to maximum possible length of the path, which is 8 for conference, 7 for Lebensmittel, 6 for Freizeit and 8 for web-directory.

Overall, our model has approximately 307K trainable parameters.

If we consider one-hop neighbours (children, datatype property neighbours and object property neighbours) as a bag, we get paths that are longer in length but fewer in number. In this case, the maximum number of paths are 2 for conference and 1 for the other 3 datasets. We take all the paths available as maximum number of paths, while for path length, we experiment from 1 to maximum length available which is 38 for conference, 16 for Lebensmittel, 12 for Freizeit and 49 for web-directory.

Another parameter denotes how we aggregate different path representations. There are two ways of doing this aggregation, by either applying a weighted sum or passing the path weights through a max-pool layer.

Package	Version
Numpy	1.18.5
Requests	2.22.0
Scipy	1.4.1
Tensorflow	2.3.0
Tensorflow-hub	0.9.0
Tensorflow-text	2.3.0
Torch	1.6.0

Table 9: Packages used and their versions

A.3 Computing Infrastructure and Training Time

We conduct all experiments on our internal GPU cluster, which runs on the Red Hat Enterprise Linux Server 7.6 (Maipo) operating system. Our CPU model is Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz and the GPU model is Nvidia Tesla K80. For all the experiments, we request 2 cores of the CPU with total memory of 40GB. The relevant software infrastructure required, including libraries and their versions used, are shown in Table 9.

With our configuration, the model took approximately 50 minutes to train on Conference dataset, 6 minutes on Lebensmittel, 10 minutes on Freizeit and 2 hours on Web Directory dataset.

Parameter	Type	Conference	Lebensmittel	Freizeit	Webdirectory	Description
Language	Fixed (As per dataset)	en	de	de	de	Language of dataset
K	Fixed (Design choice)	7	5	5	5	Value of K used in K-fold sliding window
ontology_split	Fixed (Design choice)	True	False	False	False	Split training data at ontology level (True) or on element level (False)
max_false_examples	Fixed parameter	150000	150000	150000	150000	Max number of false (dissimilar) examples to take while training
has_spellcheck	Fixed parameter	True	False	False	False	Whether or not to use an English spelling checker while preprocessing.
lr	Fixed parameter	0.001	0.001	0.001	0.001	Learning rate
num_epochs	Fixed parameter	50	50	50	50	Number of epochs
weight_decay	Fixed parameter	0.001	0.001	0.001	0.001	Weight decay
batch_size	Fixed parameter	32	32	32	32	Batch size
max_paths	Var hyperparameter	5 (3-6)	6 (1-16)	1 (1-12)	1 (1-49)	Max number of paths to consider
max_pathlen	Var hyperparameter	6 (2-26)	5 (1-7)	4 (1-6)	5 (1-8)	Max length of the path to consider
bag_of_neighbours	Var hyperparameter	False (True, False)	False (True, False)	False (True, False)	True (True, False)	Determines whether one-hop neighbors are bagged or considered as path of length 1
weighted_sum	Var hyperparameter	False (True, False)	False (True, False)	False (True, False)	False (True, False)	Determines whether unified path is obtained using weighted sum, or max pooling

Table 8: Hyperparameter chart displaying optimal values chosen. Parantheses indicate range of values tried