SORBET: a Siamese Network for Ontology Embeddings Using a Distance-based Regression Loss and BERT

Francis Gosselin and Amal Zouaq

LAMA-WeST Lab**, Departement of Computer Engineering and Software Engineering, Polytechnique Montreal, 2500 Chem. de Polytechnique, Montréal, QC H3T 1J4, Canada
{francis.gosselin,amal.zouaq}@polymtl.ca

Abstract. Ontology embedding methods have been popular in recent years, especially when it comes to representation learning algorithms for solving ontology-related tasks. Despite the impact of large language models on knowledge graphs' related tasks, there has been less focus on adapting these models to construct ontology embeddings that are both semantically relevant and faithful to the ontological structure. In this paper, we present a novel ontology embedding method that encodes ontology classes into a pre-trained SBERT through random walks and then fine-tunes the embeddings using a distance-based regression loss. We benchmark our algorithm on four different datasets across two tasks and show the impact of transfer learning and our distance-based loss on the quality of the embeddings. Our results show that SORBET outperform state-of-the-art ontology embedding techniques for the performed tasks.

Keywords: Ontology · Ontology Embedding · Transfer Learning · Representation Learning · BERT · Sentence BERT

1 Introduction

In recent years, the field of Semantic Web has been changed by the rapidly growing techniques in representation learning [19, 28]. Ontology-related tasks, such as ontology alignment or subsumption prediction, have seen an emergence of representation learning methods that have laid the foundation of future research in those fields [28]. However, there has been less focus on one of the main components of these methods: ontology embeddings. More precisely, the mapping of classes and properties of an ontology into vector representations. As representation learning gains popularity across ontology-related task, the necessity of more accurate and significant ontology embeddings is also growing. And with the rapid development of large language model that constantly outperforms state-of-the-art, such as SentenceBERT[24], there is no doubt that leveraging the

** http://www.labowest.ca/

extensive knowledge learnt by these models could be beneficial for the embedding of ontologies.

Traditional ontology embedding techniques have mainly been adapted from knowledge graph embeddings, after the known successes of the latter [12] for instance representation. TransE and TransR are good examples of well-known KG embeddings that were adapted for ontology embeddings with algorithms like DeepWalk [23] or Deep Graph Kernels [32]. However, applying the same methods and principles for ontologies and KG may not be the most adequate methodology. In comparison with KGs, ontologies contain taxonomical relations (*rdfs:subclassOf*), determining a hierarchy of parent and child concepts. KG only contains instances of those concepts, which are linked together with specific object properties. That is why rdfs:subclass relationships should be at the core of ontology embeddings methods to construct meaningful embeddings.

In this paper, we present a Siamese network for Ontology embeddings Using a Distance-based Regression Loss and BERT (SORBET), a novel ontology embedding approach. Our model is inspired by the task of ontology alignment, which learns representations of classes by bringing equivalent classes closer to each other and by pushing away the rest of the classes. We use the task of ontology alignment to obtain embeddings that are useful to represent classes and usable in other downstream tasks. More precisely, our model is inspired from the SEBMatcher system presented at OAEI 2022 [10], where BERT embeddings created from random walks based on the ontological structure was first introduced. However, SORBET changes the paradigm of the learning objective. Instead of learning to classify pairs of positive (aligned classes) and negative samples (unaligned classes), the model is asked: what should be the distance between these two classes if they were in the same ontology? This impacts the learning in a major way. First, it is flexible in that it can be used in an unsupervised or semi-supervised way. Secondly, the construction of the embeddings is driven by the structure of the ontology, leading to a major impact on their quality. Finally, as we show in the results, the training and embedding is not bound to be done on one ontology at a time, meaning SORBET can be trained on many ontologies simultaneously and transfer knowledge between them.

The contributions of this paper are summarized as follows:

- A novel ontology embedding method able to represent one or multiple ontologies in a same latent space using a pre-trained language model upon random walks. Extending our SEBMatcher model [10], SORBET is able to produce ontology embeddings in a more efficient process with a light-weight model, while yielding more accurate representations.
- A novel training objective function that injects the structure of an ontology into the latent space by reducing the distance of neighbouring classes with a regression loss.
- An improved data sampling mechanism that increases non-trivial alignment samples with an added semi-negative sampling based on graph neighbourhood.

2 Related Work

2.1 Ontology and Knowledge Graph Embeddings.

In ontology embedding, the embedding of instances, known as KG embedding, has been a popular line of research in recent years [19]. KG embedding has usually been an inspiration in the development of ontology embedding later on [18, 26]. Some popular approaches in KG embedding focus on minimizing the loss in a classification task, using correct and incorrect triples as samples from the KG. This category includes the translation-based models TransR and TransE [4, 20]. Other methods use a word embedding approach, which revolves around finding some way to express KGs in natural language sentences, then uses a NLP embedding algorithm such as a Word2vec skip-gram or CBOW [22]. One of the first methods in this category was Node2vec [11], where the main idea was to create Random Walks through the KG that would be interpreted as sentences to train the word2vec model. DeepWalk [23] and Deep Graph Kernel [32] had very similar ideas but were geared towards the analysis of social network datasets like BlogCatalog, Flickr, and YouTube. Deep Graph Kernel then extended the idea of Deep Walk by modeling graph substructures instead of Random Walks. Similarly, RDF2Vec [25] was introduced as a way to embed RDF graphs into vectors, by also using random walks, and it has proved to be effective on large datasets such as DBpedia. Finally, many recent models like KEPLER [30], K-BERT [21] and CoLAKE [29] have shown how Large Language Models (LLM) can be effective for KG embedding by producing text-enhanced entity representations.

In the field of ontology embedding, many works have been done related to the word embedding approach. Onto2Vec [26] uses a reasoner combined with the axioms of ontologies to create training data for the modeling of a word2vec skip-gram. OPA2Vec [27] extends Onto2Vec by adding the meta-data information provided by an ontology such as *rdfs:comment*. El Embeddings [18], on the other hand, expands TransE for ontologies by transforming axioms into custom losses depending on the axiom type, but does not include other ontology specific information such as meta-data. OWL2Vec* [5] takes full leverage of OWL ontologies by using ontological structure and metadata as well as a reasoner to infer axioms. It blends random walks and lexical information to fine-tune a pre-trained word2vec model.

These approaches however share some limitations. Firstly, they construct embeddings that are not meant to be generalized and likewise, knowledge is not meant to be shared across embeddings of different ontologies. Secondly, they do not leverage state-of-the-art language models, which can provide significant comprehension and depth through transfer learning.

2.2 Tasks related to ontology embeddings.

One of the main tasks related to ontologies is ontolgy alignment (OA). The ontology alignment task can be defined mathematically as the problem of finding

a mapping of semantically equivalent classes, properties or instances between two or more ontologies.

Many representation learning systems have emerged in OA in recent years. Some approaches opted for the usage of large language models as the cornerstone of their embeddings. Tom[17] and Fine-Tom [15] are both systems using the pre-trained Sentence BERT. Fine-Tom extends Tom by fine-tuning the model on the OAEI datasets. DAEOM uses BERT as part of a complex system complementing its Graph Attention Transformers (GAT). Other approaches use BERT to produce a similarity score for a pair of candidate alignment. BERTMAP [13] uses random walks to add context to the concepts and outputs a similarity score for a subset of candidate mappings. SEBMatcher [10] uses both of the methods, by leveraging Random Walks to calculate candidate alignments with BERT embeddings then doing a more accurate scoring of the pairs with a fine-tuned BERT. Other approaches using standard word embeddings have also been explored. The usage of Universal Sentence Embedding (USE) have produced good results for VeeAlign [14] and GraphMatcher [9], which are models that use path and node attention to create contextualized embeddings. LogMap-ML [6] uses OWL2Vec* embeddings fine-tuned with a supervised classification task, then use LogMap's output as anchor mappings. Finally, SCBOW+DAE [16] is a top state-of-the-art method that fine-tunes word2vec embeddings with extended knowledge coming from ConceptNet, BabelNet and WikiSynonyms. For misalignment detection, SCBOW+DAE uses a Denoising Auto Encoder (DAE) that encodes the embeddings in a smaller vector space.

Even with the success of representation learning in ontology alignment, it often face challenges when it comes to training data. Firstly, reference alignments cannot be utilized during training, or in some cases, only a small portion of such data is available. Consequently, the efficacy of systems relies heavily on the quantity of pseudo-alignments that can be generated. This challenge is amplified in cases where ontologies are small and contain minimal metadata, such as the conference track in the OAEI [2]. To overcome this hurdle, a strategy often adopted is to rely more on pre-trained word vectors, as little training can be performed. Secondly, the quality of generated training data can be poor depending of the dataset, since the high-precision positive alignments are often trivial to align. When most of the training data comprises trivial alignments, the algorithm's performance may be biased towards classifying non-trivial alignments as negative.

SORBET embeddings are partly inspired by these limitations. Indeed, it has been found that transfer learning yields state-of-the-art results [13, 31], however, the quality of the embeddings can still be poor due to the mentioned problems that arise for OA. This motivates the idea of an ontology embedding technique based on transfer learning that is not bound to the training objective of traditional OA models.

3 Methodology

The foremost idea behind SORBET is to create BERT embeddings that are representative of the ontological structure. To achieve this, close pairs of classes in the ontology must be pushed together while distinct classes must be pushed appart. Hence, a siamese network architecture is employed.

3.1 Architecture.

SORBET Embeddings follow a siamese architecture pattern using Sentence BERT[24], a pre-trained Siamese BERT model. Figure 1 shows the general architecture of our model. In the SentenceBERT paper, the authors fine-tuned the model on a binary classification task where pairs of sentences are either classified as synonyms or antonyms. SORBET follows the same principle, but instead of a pair of sentences, a pair of tree walks representing classes is fed to the model. The output is then filtered by the pooling layer and a regression distance-based loss is applied. The tree walks are obtained through our data sampling mechanism, which generates positive, semi-negative and negative samples in a stochastic process.

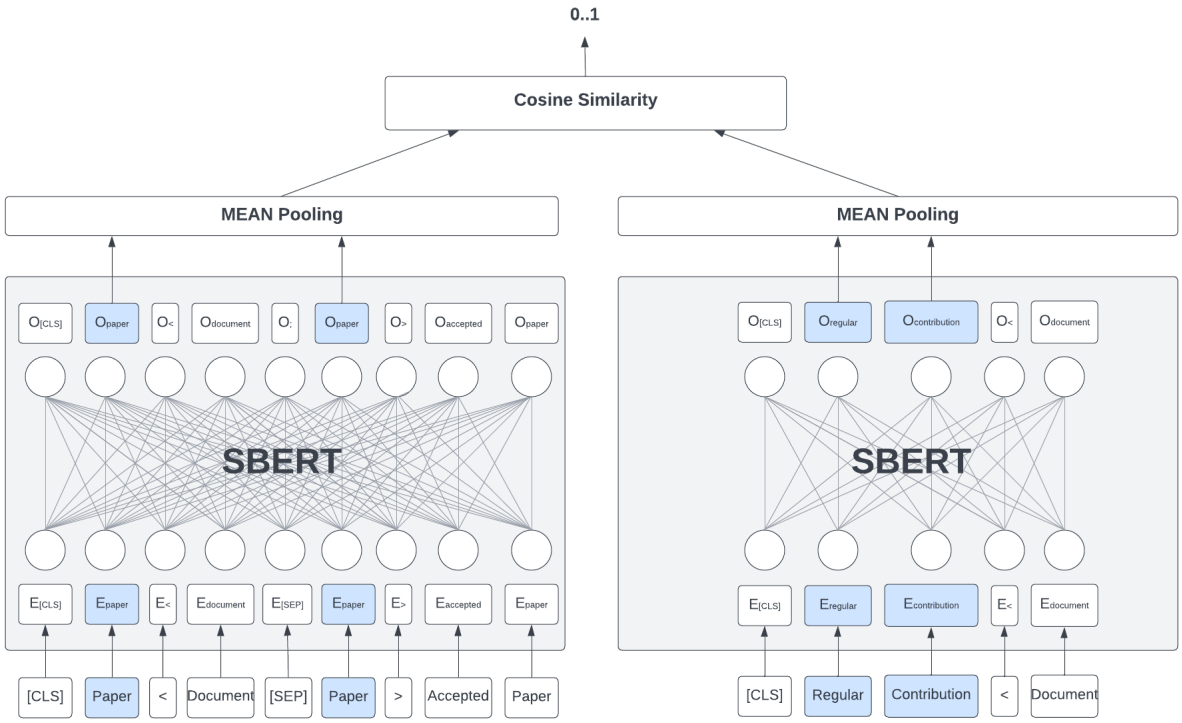


Fig. 1: General Architecture

Preprocessing For every class and property, the associated descriptive label is tokenized and case-folded. If the tokens are not part of the pre-trained BERT vocabulary, a basic spelling corrector algorithm is applied[1] to deal with some errors in input labels. For ontologies that associate more than one label or synonyms to classes, SORBET creates a list of descriptive terms for these classes and randomly chooses labels from this list during training.

Tree Walk A tree walk is the process of finding multiple distinct random walks, where a random walk is a sequence of classes, starting from a given concept and iterating randomly through the ontological structure. Each iteration adds a subsequent class randomly chosen from the set of 1-hop neighbors of its preceding class in the random walk, by considering subclassOf and object properties relations. These random walks are used as the textual representation of the root concept and its context when they are fed into the BERT model. The algorithm of the Tree Walk, algorithm 1, is a derivation of the original algorithm described in SEBMatcher [10]. The first change is that both the number of tree walks and the walk length are now interval hyper-parameters instead of having a fixed minimum of 0, so to increase regularization. Furthermore, the concatenation token was changed to "[SEP]" instead of ";". Experiments showed that using the pre-trained special token "[SEP]" produced slightly better results than the previous approach.

Algorithm 1 Tree walk

Input: *Source ontology O, concept* c_0 *from the ontology O*
Output: T

Initialize set of visited nodes: $C_v \leftarrow \{c_0\}$
Initialize tree walk: $T \leftarrow [\,]$
$n\_branch \leftarrow randint(1..number\ of\ neighbours)$
for $i := 1$ *to* n_branch **do**
    Initialize walk: $W \leftarrow [\,]$
    $walk\_len \leftarrow randint(1..max\_len)$
    Append c_0 *to* W
    for $j := 1$ *to* $walk\_len$ **do**
        $current\_concept \leftarrow W_j$
        $neighbours \leftarrow get\_neighbours(current\_concept)\ /C_v$
        $next\_concept,\ relation\ \leftarrow choose\_random\_neighbour(neighbours)$
        Append relation to W
        Append $next\_concept$ *to* W
        $C_v \leftarrow C_v \cup next\_concept$
    Append END_TOKEN *to* W
    $T \leftarrow T + W$

[1] https://github.com/filyp/autocorrect

Random Masks A masking strategy is used during training with the idea of regularizing the model. In fact, an undesirable effect during training would be that the model overfits on the label of the root concepts alone without using the context provided by the Tree Walk. In order to prevent this, the root concept is entirely masked 15% of the time, while the remaining 85% of samples will have a random mask applied to 15% of the subtokens.

MEAN Pooling The pooling layer intends to compute the final vector representation of a concept derived from the output of the BERT model. To do so, the pooling layer computes the mean vector of the BERT outputs related to the root concept (highlighted in figure 1). Subsequently, all other embeddings in the tree walk are discarded.

3.2 Data Sampling.

Training data consists of pairs of classes that should be pushed toward or apart from each other depending on their similarity score (1 meaning they should be pushed together). It is composed of positive, semi-negative and negative samples. Positive samples are pairs of classes that have a similarity score of 1, meaning they are equivalent concepts. Semi-negative samples are pairs of distinct neighbouring classes in the ontological structure, meaning their similarity score is between 0 and 1 exclusively. Negative samples are distinct classes with disjoint neighbourhood, they have a similarity score of 0. The training data is obtained through our data sampling mechanism, which generates pairs of concepts in a stochastic process using two different sampling strategies: intra-ontology sampling and inter-ontology sampling. Intra-ontology sampling refers to pairs of concepts from the same ontology while inter-ontology sampling refers to pairs of concepts from two different ontologies. Inter-ontology sampling can only be applied if the two following conditions are met: there are 2 or more ontologies, and positive alignments can be inferred from these ontologies. The utilisation of both strategies enhances data augmentation and makes SORBET flexible for the embedding of one or several ontologies.

Positive sampling Firstly, we create a set of positive intra-ontology samples where each concept is paired with itself $M^{+}_{intra} : \{(c_i, c_i) : \forall c_i \in O, O'\}$. Then a set of inter-ontology samples is obtained through a String matcher, which is the simple process of matching each concept from a source and target ontology that has the exact same rdfs:label. To further augment the quality of training data, we sample a subset of positive alignments that are chosen randomly from the reference alignments of the ontology alignment task. The resulting set of alignments denoted as M^{+}_{inter} is the union of the String matched samples and the reference alignments samples.

Semi-negative sampling Semi-negative samples use both intra and inter-ontology sampling strategies. A semi-negative sample is obtained by choosing

a random concept c_i from the ontology O and pairing it with any another concept c_j from either the same ontology O or a different ontology O' such that $d(c_i, c_j) < A$. The definition of the distance function d and the hyper-parameter A are described in section 3.3.

Negative sampling Negative samples are generated much like semi-negative samples, but with the opposite condition $d(c_i, c_j) \geq A$. Additionally, cross-ontology negative pairs obtained with the inter-ontology strategy cannot have an undefined distance between them. Such case would happen if the cross-ontology distance between the pair cannot be approximated, for example if there are no positive sample $(c_i, c_j) \in M^+_{inter}$ such as $c_i \in O, c_j \in O'$.

The final training set M is built by carefully balancing the generation of samples. For every learning batch, the ratio of positive samples is set to $\lambda_{positive}$, the ratio of semi-negative to negative samples is λ_{semi} and finally the ratio of the usage of inter to intra sampling strategy is λ_{inter}. During the training, one epoch corresponds to one iteration of the model on all positive samples and for each batch, new semi-negative and negative samples are generated. It is necessary for the process to be stochastic, since there are too many possibilities of semi-negative and negative pairs to realistically iterate through them all.

3.3 Regression Distance-based Loss.

The Regression Distance-based Loss is a geometric approach that is inspired by the traditional classification losses used in ontology alignment. The objective of this function is to calculate a similarity score from a pair of classes that is proportionate to the distance between them. We compare this similarity score with the cosine similarity generated by the Siamese network using Mean Squared Error (MSE), and use this metrics for the gradient descent. The equation of the loss can be defined as the following:

$$L = \frac{1}{|M|} \sum_{(c_i, c_j) \in M} [(sim_\theta(c_i, c_j) - \frac{A - min(d(c_i, c_j), A)}{A})^2] \quad (1)$$

Where the function sim_θ is the cosine similarity of the vector representation of the concepts produced by the Siamese network. A is an upper bound on the distance between concepts, its optimal value can vary across datasets and can be tuned as a hyper-parameter. The function d is defined in the following section.

Definition 1. *Let d be a function returning the shortest distance between a source concept c_i and a target concept c_j. Where a distance can be calculated by iterating through the ontological structure only considering rdfs:subclassof relationships.*

SORBET uses a **cross-ontology distance**, which is a distance where the source c_i and target concept c_k originate from different ontologies. Since there are no direct connections between 2 different ontologies, a distance cannot be

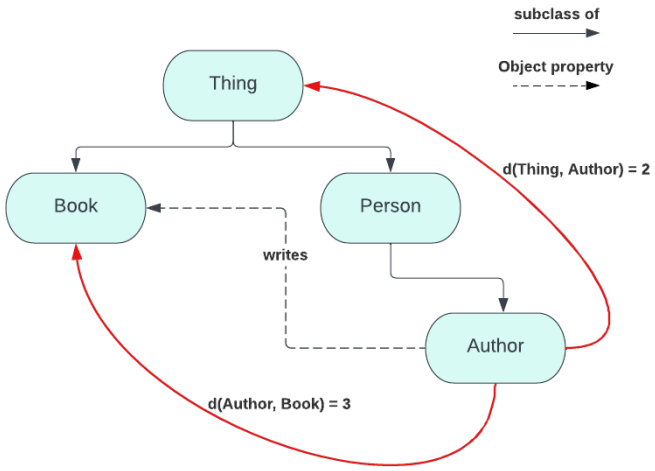


Fig. 2: Example in which the distance between Author and Person in the source Ontology is 1, while the distance between Author and Book is 3.

fetched directly. However we can utilize a positive mapping $(c_i, c_j) \in M^+_{inter}$ to approximate the distance $d(c_i, c_j)$ to $d(c_j, c_k)$ instead.

$$d(c_i, c_k) \approx d(c_j, c_k | (c_i, c_j) \in M^+_{inter}) \tag{2}$$

where $c_i \in O$ and $c_j, c_k \in O'$

The distance function can be visualized in both figure 2 and figure 3. In figure 2, the path between Author and Thing is obtained with two iterations through the hierarchical structure, it can also be observed that the distance between Author and Book is three even if they are linked by an object property. In figure 3, the knowledge of the alignment of Person and Human allows the model to approximate the distance between Book and Human.

4 Experiments

We experiment with SORBET embeddings on two downstream tasks: ontology alignment and ontology subsumption.

Formally, given two ontologies O and O', the ontology alignment task can be defined as finding a mapping M between the set of concepts C in O and the concepts C' in O', such that M is a subset of CxC', and each pair of concepts in M is a pair of equivalent concepts.

The subsumption prediction task is a task that involves predicting whether one concept in an ontology O is a subclass of another concept in the same or another ontology. It is a classification problem, where the input is a pair of concepts (A, B), and the output is a binary label indicating whether A is subsumed by B or not.

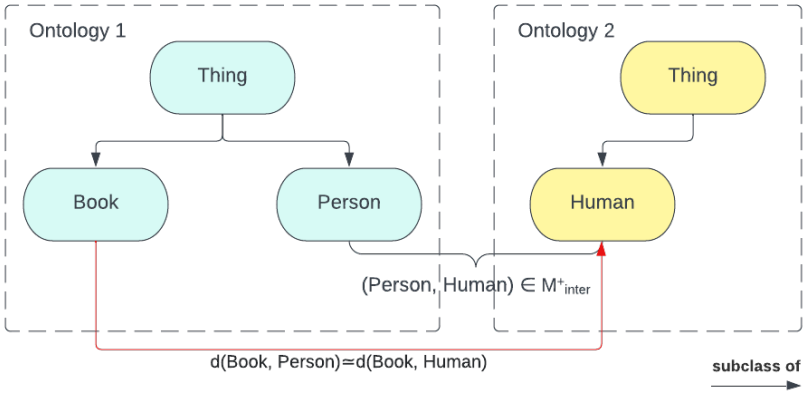


Fig. 3: Example of the approximation of a distance between cross-ontology concepts

4.1 Datasets.

Table 1 shows some descriptive statistics on our datasets.

For the evaluation on the ontology alignment task, the OAEI 2022 [2] tracks Anatomy and Conference were used as the evaluation datasets. The Conference track is a combination of 16 small ontologies totaling 867 classes describing the conference organization domain. The evaluation is performed with the reference alignments of a subset of 7 of those ontologies, using the *ra1-m1* subset. The Anatomy track is the alignment of the mouse anatomy (MA) and the human anatomy (NCI), with respectively 2744 and 3304 classes. These tracks were trained and evaluated simultaneously with the same SORBET model.

For the subsumption task, our model is benchmarked on two different datasets: FoodOn[8] and Gene Ontology (GO)[3, 7] . FoodOn[3] is an ontology capturing a vast amount of information about food, it has 28,182 classes and 29,778 subsumption relations. GO[4] is a well-known bioinformatic ontology, that captures information about the functions of genes, with 44,244 classes and 72,601 subsumption relations.

4.2 Model.

The SORBET model used in the experiments is fine-tuned from the pre-trained Sentence BERT weights trained on the natural language inference (NLI) dataset, it is available in Huggingface using *sentence-transformers/bert-base-nli-mean-tokens*. This version of SentenceBERT was used since it has the advantage of being already pre-trained on the similar task of scoring similar inputs.

[2] https://oaei.ontologymatching.org/2022/
[3] https://foodon.org/
[4] http://geneontology.org/ontology/ accessed on the 2020-09-08

Table 1: Statistics of the benchmarked datasets and their ontologies

Datasets	Number of ontologies	Avg. number of classes	Avg. number of subsumptions
Conference	16	54	78
Anatomy	2	3,024	4,958
FoodOn	1	28,182	29,778
GO	1	44,244	72, 601

Hyperparameters. For the loss function, we set the upper maximum bound $A = 4$. In the alignment sampling, we set $\lambda_{positive} = 0.5$, $\lambda_{semi} = 0.6$, $\lambda_{inter} = 0.8$, and 20% of the available reference alignments in the tracks are used in training. Moreover, the set of considered neighbours in the construction of the Random Walks for contextual data is composed of parents, children and object properties. The impacts of those hyper-parameter are kept for future works. Each epoch iterates over all positive alignments and the batch size is set to 32. The training lasts for 3 epochs. During both training and prediction, the number of paths in the tree walk is randomly chosen in the interval $[0, 5]$, and the length of each path in the interval $[2, 6]$.

4.3 Experiments on the Ontology Alignment Task

In this section, SORBET embeddings, as well as other baseline embeddings, are evaluated on the ontology alignment task. To achieve this, we use the embeddings obtained through various methods to compute the similarity of pairs of concepts, and sort them in decreasing order. Then for each reference alignment, we evaluate how far off the pair was in the ordered list of predicted pairs. Since our model is trained on part of the reference alignments, the testing dataset excludes all of the training alignments. Our evaluation metric is Hits@K:

$$Hits@K = \frac{|\{m \in M_{ref}/M^+_{ref} | Rank(m) \leq K\}|}{|M_{ref}/M^+_{ref}|} \tag{3}$$

Where M_{ref} is the set of reference alignments, M^+_{ref} is the subset (20%) of the reference alignments used in training. The function $Rank()$ is a function that determines how far off the algorithm was to giving the right alignment. To achieve this, it outputs the rank of the reference alignment (c_i, c_j) the list of predicted pairs of alignments $[(c_i, c_0), (c_i, c_1), ...(c_i, c_j)...(c_i, c_n)]$ where pairs are listed in descending order according to the cosine similarity between the two embeddings produced by the model in table 2.

Table 2 presents the performance of SORBET embeddings for the ontology alignment tasks on the anatomy and conference tracks of the OAEI 2022. For this task, we compare our model with the baseline OWL2Vec*, a state-of-the-art model in ontology embeddings. We also compare our model with SentenceBERT, a state-of-the-art sentence embedding model, for which the embeddings of classes

are fetched by passing their preprocessed label in the model. OWL2Vec* embeddings were obtained using the code provided by its authors[5], while the SentenceBERT model can be obtained through its main version on *huggingface*. [6]. Another aspect of this study is the comparison of regression distance-based loss (SORBET) to the traditional classification loss used in OA models. Therefore, we ran another version of SORBET for which the regression distanced-based loss is replaced with a classification loss (SORBET(Classification)). This is done by omitting the creation of semi-negative samples ($\lambda_{semi} = 0.0$).

Table 2: Comparison of different embeddings for the ontology alignment task

Models	Conference			Anatomy		
	Hits@1	Hits@5	Hits@10	Hits@1	Hits@5	Hits@10
SORBET(Classification)	0.8108	0.9034	0.9305	0.8502	0.9307	0.9426
SentenceBert	0.7568	0.8687	0.8996	0.7784	0.8740	0.8984
Owl2Vec*	0.7876	0.9073	0.9343	0.7427	0.8232	0.8470
SORBET Embeddings	**0.9095**	**0.9809**	**0.9904**	**0.9024**	**0.9636**	**0.9760**

As the result shows, SORBET outperforms the baselines for all metrics. It can also be noticed that the performance is not affected by the training of SORBET on different domains and ontology sizes simultaneously.

4.4 Experiments on the Subsumption Task.

Table 3 shows the performance of SORBET embeddings on the subsumption task using the FoodOn and GO case studies. For this task, the employed baselines are: RDF2Vec [25], a well-known KG embedding algorithm, Onto2Vec[26], which is a more traditional ontology embeddings model, OWL2Vec*, SentenceBert and SORBET(classification). The performance of baselines models are taken from the benchmark of the Owl2Vec* paper [5]. The following Hits@K is computed, where M_{sub} is a test set of subclass axioms that were removed from the ontology used in training.

$$Hits@K = \frac{|\{m \in M_{sub} | Rank(m) \leq K\}|}{|M_{sub}|} \tag{4}$$

The $Rank()$ function is similar to the one in the OA task, however, the similarity score is changed to follow the same methodology of the OWL2Vec* paper [5]. In this evaluation framework, the embeddings of every concept is obtained using one of the models in Table 3. Then, using those embeddings, a Random Forest classifier is trained to classify a dataset of true and false subclass relationship pairs. Finally, for every pair of concepts, instead of the cosine similarity,

[5] https://github.com/KRR-Oxford/OWL2Vec-Star
[6] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

their embeddings are fed to the Random Forest classifier and the output is used as the similarity score.

Table 3: Comparison of different embeddings for the subsumption prediction task

Models	FoodOn			GO		
	Hits@1	Hits@5	Hits@10	Hits@1	Hits@5	Hits@10
RDF2Vec	0.053	0.097	0.119	0.017	0.057	0.087
Onto2Vec	0.014	0.047	0.064	0.008	0.031	0.053
OWL2Vec*	0.143	0.287	0.357	0.076	0.258	0.376
SentenceBERT	0.074	0.186	0.256	0.059	0.171	0.225
SORBET (classification)	0.040	0.060	0.080	0.039	0.120	0.158
SORBET embeddings	**0.169**	**0.417**	**0.521**	**0.090**	**0.310**	**0.423**

Overall, SORBET embeddings outperform the state-of-the-art for both the ontology alignment and subsumption tasks. Without surprise, the task of OA has the most noticeable difference. Not only because the training objective is partly a OA task objective, but also because the conference dataset does not gather much training data, meaning transfer learning has the uttermost importance. It is also noticeable that while the results are high, no hyper-parameter tuning is done for the different datasets and tasks, therefore showing the regularization of the model. Finally, as the same embeddings were all learnt and utilized simultaneously for the benchmark of the datasets, the model is able to generalize while preventing overfitting on a single ontology.

4.5 Ablation Study.

In this ablation study, we measure how specific components of the model have an impact on the final result.

The pre-training p of the BERT refers to the usage of the pre-trained Sentence BERT. w indicates whether the model used Tree Walks in training, if not, only the label of concepts are used as input. Finally, r indicates if a portion (20%) of the reference alignments were used as positive alignments.

The results of the ablation study, table 4, demonstrate the importance of the different aspects of SORBET embeddings. First of all, the regression loss makes most of the boost in performance in both tested datasets compared to the traditional classification loss. The augmented input with context had a large impact for the conference dataset, however the same cannot be said for the anatomy track. This could be because of the difference in the nature of both datasets. In fact, the labels of concepts in the conference dataset tend to be less detailed, which may be why the use of the context makes such difference. Finally, as expected, the use of reference alignments in training increases the performance by a significant margin. Even though the Hits@5 and Hits@10 do

not seem impacted by this change, the small performance gap could be due to the already very high scores.

Table 4: Results of the ablation study

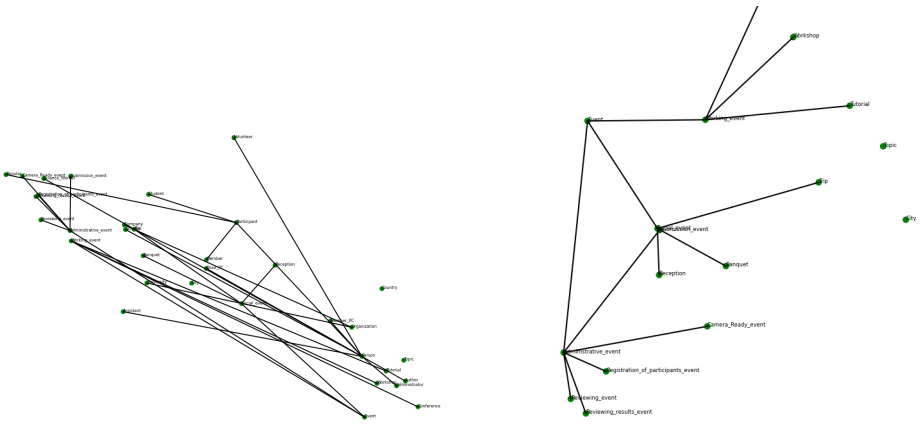
Models	Conference			Anatomy		
	Hits@1	Hits@5	Hits@10	Hits@1	Hits@5	Hits@10
SORBET	0.7374	0.8996	0.9112	0.7790	0.8760	0.9010
SORBET$_p$	0.8069	0.9537	0.9730	0.8845	0.9512	0.9611
SORBET$_{p+w}$	0.8764	0.9652	0.9846	0.8812	0.9538	0.9690
SORBET$_{p+w+r}$(Classification)	0.8108	0.9034	0.9305	0.8502	0.9307	0.9426
SORBET$_{p+w+r}$	**0.9095**	**0.9809**	**0.9904**	**0.9024**	**0.9636**	**0.9760**

5 Discussion

5.1 Ontological representation.

The main goal of SORBET embeddings is to obtain a more accurate representation of the ontological structure in the latent space. This goal is achieved by tweaking a OA classification loss so that every neighbour concepts have a gradually higher loss the further they are apart from each other. This constraint inherently builds the latent space in the desired way because the embeddings that would produce the minimum possible loss are the ones where the ontological structure could be perfectly deduced from the latent space. One could view this as a web of concepts being held together by subclassOf relationships. Conversely, with a classification loss, a model does not have any way of keeping neighbouring concepts together in a structured way. Equivalent concepts are indeed pushed together, however, negative samples push every other pair of concepts away from each other. This results into a very chaotic latent space.

The experiment in figure 4 demonstrates this phenomenon. We initially plot the latent space of the embeddings created by SORBET into a 2D space using PCA for a single ontology. Secondly, we plot the same ontology but with embeddings resulting from SORBET trained with a classification loss. We then plot every rdfs:subclass relationship in order to visualise the structure of the ontology in the latent space. The results show undoubtedly that the regression loss creates a more organised space. The tree-like structure depicted in the projection imitates the hierarchical structure of the ontology: the leaf nodes aggregate into a larger group of nodes which then aggregate to the root (top-level class). Furthermore, the model creates separate clusters for different aspects of the ontology. The cluster on the left represent every element that is a derivative of "Event", while the ontology on the right regroups all derivatives of "Person". Isolated concepts are separated from any of these clusters.



(a) Classification Loss embeddings (b) Regression Loss embeddings

Fig. 4: Comparison of embeddings obtained with different loss functions, using the ontology "confof" in the Conference track

5.2 Superposed ontologies.

An important feature of SORBET embeddings is that the concepts of any ontologies can be trained and superposed in the same latent space. As the previous analysis showed, the nature of the training objective leads to clusters of tree-like structures for each distinct ontologies to be distributed across the vector space. But the training objective of the regression loss is also directed at making overlaps of similar clusters. For example, a cluster of the ontology O associated with the concept "Person", and another cluster from the ontology O' associated with concept "User" should have an overlap. This can be achieved in our loss function with inter-ontology positive and negative sampling if there is at least one mapping that anchors the clusters. In this example, if "User" and "Person" both have the child concept "Member", the two clusters will be pushed towards each other, resulting in an overlap. Consequently, the embeddings are much more coherent, and this enables the possibility of training on multiple domains simultaneously.

5.3 Limitations of our approach

While SORBET has many perks, there are still drawbacks to our approach. The most evident one is that, in order to construct training data, there must be a high amount of *rdfs:subclassOf* relationships. In fact, the more an ontology has an average depth closer to 0, the more the learning objective becomes a classification task. Consequently, the usage of the regression loss become much more advantageous when trained on deep and complex ontologies. Another flaw of the approach is the immutable value of a distance between 2 concepts related by a single $rdfs : subclassOf$ relationship. The underlying fact for this hypothesis is

that concepts closer to the root of an ontology have different taxonomic relations than concepts further down the hierarchy. For example, the distance between the pair "Thing" and "Person" should be higher than the distance between "Paper" and "Short Paper", even though both pairs are neighbouring concepts.

6 Conclusion

In this paper, we presented how SORBET embeddings are a better alternative to traditional classification-refined embeddings. The Hits@K have shown a significant improvement, indicating a higher quality of the embeddings. The visual analysis also demonstrates the continuity of the ontology's structure into the latent space, and how this affects the performance of the model. In addition SORBET embeddings tend to be much less chaotic than those obtained using classification, yielding more robust results. In future work, we plan to experiment with different combinations of rules that could improve the estimation of distances between concepts, as well as finding new ways to train our model on shallow ontologies.

Supplemental Material Statement Source code and datasets can be found on its Github repository[7]

Acknowledgment This research has been funded by the NSERC Discovery Grant Program. The authors acknowledge support from Compute Canada for providing computational resources.

References

1. Ontology Matching 2021 : Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), CEUR Workshop Proceedings, vol. 3063. CEUR-WS.org (2021)
2. Ontology Matching 2022 : Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, virtual conference, October 23, 2022, CEUR Workshops Proceedings, vol. 3324. CEUR-WS.org (2022)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. Nature Genetics **25**(1), 25–29 (May 2000). https://doi.org/10.1038/75556, https://doi.org/10.1038/75556
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)

[7] https://github.com/Lama-West/SORBET_ISWC23

5. Chen, J., Hu, P., Jimenez-Ruiz, E., Holter, O.M., Antonyrajah, D., Horrocks, I.: OWL2Vec*: embedding of OWL ontologies. Machine Learning **110**(7), 1813–1845 (2021). https://doi.org/10.1007/s10994-021-05997-6
6. Chen, J., Jiménez-Ruiz, E., Horrocks, I., Antonyrajah, D., Hadian, A., Lee, J.: Augmenting ontology alignment by semantic embedding and distant supervision. In: The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18. pp. 392–408. Springer (2021)
7. Consortium, T.G.O.: The Gene Ontology resource: enriching a GOld mine. Nucleic Acids Research **49**(D1), D325–D334 (12 2020). https://doi.org/10.1093/nar/gkaa1113, https://doi.org/10.1093/nar/gkaa1113
8. Dooley, D.M., Griffiths, E.J., Gosal, G.S., Buttigieg, P.L., Hoehndorf, R., Lange, M.C., Schriml, L.M., Brinkman, F.S.L., Hsiao, W.W.L.: Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Science of Food **2**(1), 23 (Dec 2018). https://doi.org/10.1038/s41538-018-0032-6, https://doi.org/10.1038/s41538-018-0032-6
9. Efeoglu, S.: Graphmatcher: A graph representation learning approach for ontology matching. In: Ontology Matching 2022 : Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, virtual conference, October 23, 2022 [2], pp. 174–180
10. Gosselin, F., Zouaq, A.: Sebmatcher results for oaei 2022. In: Ontology Matching 2022 : Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, virtual conference, October 23, 2022 [2], pp. 202–209
11. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
12. Gutiérrez-Basulto, V., Schockaert, S.: From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In: International Conference on Principles of Knowledge Representation and Reasoning (2018)
13. He, Y., Chen, J., Antonyrajah, D., Horrocks, I.: Bertmap: a bert-based ontology alignment system. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 5684–5691 (2022)
14. Iyer, V., Agarwal, A., Kumar, H.: VeeAlign: Multifaceted context representation using dual attention for ontology alignment. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 10780–10792. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.842, https://aclanthology.org/2021.emnlp-main.842
15. Knorr, L., Portisch, J.: Fine-tom matcher results for oaei 2021. In: Ontology Matching 2021 : Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021) [1], pp. 144–151
16. Kolyvakis, P., Kalousis, A., Smith, B., Kiritsis, D.: Biomedical ontology alignment: an approach based on representation learning. Journal of biomedical semantics **9**(1), 1–20 (2018)
17. Kossack, D., Borg, N., Knorr, L., Portisch, J.: Tom matcher results for oaei 2021. In: Ontology Matching 2021 : Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021) [1], pp. 193–198

18. Kulmanov, M., Liu-Wei, W., Yan, Y., Hoehndorf, R.: El embeddings: Geometric construction of models for the description logic el ++. In: International Joint Conference on Artificial Intelligence (2019)
19. Li, C., Li, A., Wang, Y., Tu, H., Song, Y.: A survey on approaches and applications of knowledge representation learning. In: 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC). pp. 312–319. IEEE (2020)
20. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
21. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-bert: Enabling language representation with knowledge graph. In: AAAI Conference on Artificial Intelligence (2019), https://api.semanticscholar.org/CorpusID:202583325
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013)
23. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
24. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing (2019)
25. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: Rdf2vec: Rdf graph embeddings and their applications. Semantic Web **10**(4), 721–752 (2019)
26. Smaili, F.Z., Gao, X., Hoehndorf, R.: Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. Bioinformatics **34**(13), i52–i60 (2018)
27. Smaili, F.Z., Gao, X., Hoehndorf, R.: Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics **35**(12), 2133–2140 (2019)
28. Sousa, G., Lima, R., Trojahn, C.: An eye on representation learning in ontology matching. In: Ontology Matching 2022 : Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, virtual conference, October 23, 2022 [2], pp. 49–60
29. Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X., Zhang, Z.: Colake: Contextualized language and knowledge embedding. ArXiv **abs/2010.00309** (2020), https://api.semanticscholar.org/CorpusID:222090412
30. Wang, X., Gao, T., Zhu, Z., Liu, Z., Li, J.Z., Tang, J.: Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics **9**, 176–194 (2019), https://api.semanticscholar.org/CorpusID:208006241
31. Wu, J., Lv, J., Guo, H., Ma, S.: Daeom: A deep attentional embedding approach for biomedical ontology matching. Applied Sciences **10**, 7909 (2020)
32. Yanardag, P., Vishwanathan, S.: Deep graph kernels. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1365–1374 (2015)