A Single Vector Is Not Enough: Taxonomy Expansion via Box Embeddings

Song Jiang
University of California, Los Angeles
Los Angeles, CA
songjiang@cs.ucla.edu

Qiyue Yao
University of California, Los Angeles
Los Angeles, CA
qiyueyao@g.ucla.edu

Qifan Wang
Meta AI
Menlo Park, CA
wqfcr@meta.com

Yizhou Sun
University of California, Los Angeles
Los Angeles, CA
yzsun@cs.ucla.edu

ABSTRACT

Taxonomies, which organize knowledge hierarchically, support various practical web applications such as product navigation in online shopping and user profile tagging on social platforms. Given the continued and rapid emergence of new entities, maintaining a comprehensive taxonomy in a timely manner through human annotation is prohibitively expensive. Therefore, expanding a taxonomy automatically with new entities is essential. Most existing methods for expanding taxonomies encode entities into vector embeddings (i.e., single points). However, we argue that vectors are insufficient to model the "is-a" hierarchy in taxonomy (*asymmetrical* relation), because two points can only represent pairwise similarity (*symmetrical* relation). To this end, we propose to project taxonomy entities into *boxes* (i.e., hyperrectangles). Two boxes can be "contained", "disjoint" and "intersecting", thus naturally representing an asymmetrical taxonomic hierarchy. Upon box embeddings, we propose a novel model BoxTaxo for taxonomy expansion. The core of BoxTaxo is to learn boxes for entities to capture their child-parent hierarchies. To achieve this, BoxTaxo optimizes the box embeddings from a joint view of geometry and probability. BoxTaxo also offers an easy and natural way for inference: examine whether the box of a given new entity is fully enclosed inside the box of a candidate parent from the existing taxonomy. Extensive experiments on two benchmarks demonstrate the effectiveness of BoxTaxo compared to vector based models.

KEYWORDS

Taxonomy, Box Embeddings, Geometry, Representation Learning

ACM Reference Format:
Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. 2023. A Single Vector Is Not Enough: Taxonomy Expansion via Box Embeddings. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3543507.3583310

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583310

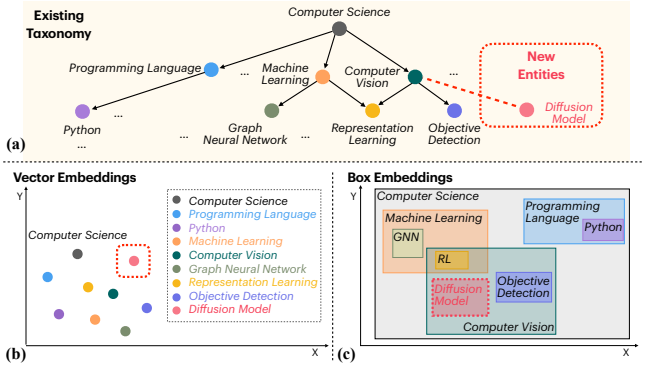

Figure 1: Comparison of taxonomy expansion using vector embeddings and box embeddings. (a) An example of taxonomy expansion. A new entity "Diffusion Model" is to be attached to its appropriate category "Computer Vision". (b) Vector (point) embeddings of taxonomy. Two points can only represent similarity (distance). (c) Box embeddings of taxonomy. Two boxes can represent the taxonomic "is-a" relation.

1 INTRODUCTION

A taxonomy is a schema of hierarchical classification, which is used to organize conceptual entities into a tree-like structure according to their semantics. Taxonomies have been widely adopted to support various web services because of the effectiveness of indexing and organizing knowledge. For example, Amazon has a product taxonomy to facilitate online shopping [22], and Pinterest uses taxonomy to enhance content understanding and recommendation [11, 21]. Many taxonomies were initially curated by domain experts, however, due to the constant and rapid growth of new concepts, automatically expanding existing taxonomies with these new entities is necessary to avoid their obsolescence. Fig. 1 (a) shows an example of taxonomy expansion: a new research direction "Diffusion Model" is appended to its category "Computer Vision", forming a child-parent hierarchy. For consistency with existing literature, we follow [33, 42] and refer to a child as *query*, and a parent as *anchor*. The terms are interchangeable throughout this paper.

Existing approaches for taxonomy expansion focus on capturing the child-parent hierarchies. Early efforts learn the hierarchies by exploiting the semantic relatedness between two entities. The semantics can be represented by lexical patterns [12, 35] or later

the more powerful distributional word embeddings [3, 9, 26]. Beyond semantics, recent works have further explicitly modeled the tree structure of taxonomy. They use various structural summaries, including paths [13, 18, 42] and local graphs [22, 33, 39], as additional signals to enhance the learning of child-parent hierarchies. Representing hierarchies is more in line with the geometric properties of hyperbolic space [10, 27]. Several works [2, 20] model the child-parent relations by learning hyperbolic representations.

The core methodology of most aforementioned approaches is to learn *vector* embeddings for entities in taxonomy. The child-parent relation is then inferred by computing the relatedness of a pair of entities upon their vector embeddings. However, the vector embeddings, i.e., points in geometric space, can only represent the pairwise similarity, which is a *symmetrical* relation (Similarity is usually measured by distance – either Euclidean or geodesic [1] – of two points). The taxonomic child-parent hierarchies, on the contrary, are naturally *asymmetrical*. Therefore, the vector based embeddings are *not* sufficient to represent the hierarchies in taxonomy, limiting their effectiveness in taxonomy expansion.

To overcome this insufficiency, instead of vectors, we propose to use *boxes* to represent the entities in the taxonomy. A box is an axis-aligned hyperrectangle in geometric space, which can be characterized by two points. Unlike a single point, the benefit of a box is that box has a geometric region, which enables it to represent the more complicated asymmetrical pairwise relations such as "enclose", "disjoint" and "intersect". Fig. 1 (b) and Fig. 1 (c) show this superiority of box embeddings over vector embeddings. Specifically, a child box is entirely enclosed inside its parent box (e.g., "Graph Neural Network" and "Machine learning"). Two entities are fully separated if they are not in a child-parent hierarchy (e.g., "Programming Language" and "Machine learning"). The boxes of two entities overlap if they share common children in taxonomy (e.g., "Computer Vision" and "Machine learning").

Despite the natural and intuitive representation of taxonomic hierarchies, the box embeddings for taxonomy expansion still face three main challenges. First, limited taxonomy annotation is available for new entities, making it difficult to learn accurate boxes and infer their positions in the taxonomy in a supervised manner. Second, most existing box embeddings approaches optimize boxes by capturing probabilistic properties, which have proven difficult to train in practice [16, 38]. The reason is box pairs that are supposed to "enclose" or "intersect", but are wrongly disjoint during training, will never be corrected because the gradients from the probabilistic loss function are zero in this case. [7, 16] mitigate this issue by representing the edges of boxes as probabilistic density distributions, i.e., making the box "soft". However, such "soft" boxes lose the intuitive interpretability of normal "hard" boxes. Third, different from reasoning in the existing structure, taxonomy expansion requires learning boxes for new entities. Therefore, a desired model should be generalizable, which is able to generate box embeddings compatible with existing taxonomies for new entities.

In this paper, we propose BoxTaxo, a self-supervised model that expands taxonomy with box embeddings. With self-supervised learning, our model does not require annotated labels, but creates training samples from the existing taxonomy. Specifically, each ⟨child, parent⟩ pair in the existing taxonomy is treated as a positive sample. The entities that are not the ancestors of each child are collected as negative samples. To optimize the box embeddings, we propose a joint loss function that guides the boxes to capture the taxonomic hierarchies from both the geometric view and the probabilistic view. The joint view loss function can avoid the gradient missing issue mentioned above and still ensure the boxes are intuitive and interpretable to humans. The box embeddings are encoded via a pre-trained language model to ensure the generalizability to new entities. At inference time, box embeddings offer an easy and natural way to find an appropriate anchor for a query, by checking whether the box of a candidate parent fully contains the box of the query. We implement this from the probabilistic view in BoxTaxo.

Our **main contributions** are summarized as follows: 1) We propose to use box embeddings for taxonomy expansion, which can accurately represent the hierarchies in taxonomy. 2) We develop a self-supervised model that optimizes the box embeddings through joint learning of geometry and probability. 3) We conduct an extensive set of experiments on two real-world taxonomies. Experimental results demonstrate the effectiveness of BoxTaxo compared to vector based representations. We also provide various ablation studies and analyses to understand how BoxTaxo works.

Scope and Limitation. This work is an early attempt to use box embeddings for representing and expanding an existing taxonomy. Our main focus is to study whether box embeddings are more suitable than single vectors for this task. We would like to keep the model as simple as we can in this step. Therefore we only model the ⟨child, parent⟩ pairs and do not utilize the complicated structural signals, such as paths [13, 18, 42] and local graphs [22, 33, 39], or check more contexts to enhance the anchor representation [40, 44]. We are aware that such advanced structures have the potential to further boost the box embeddings learning and thus improve the taxonomy expansion task. However, how to facilitate box embeddings with structure signals is out of the scope of this paper. We hope this work can inspire future studies in these directions.

2 RELATED WORK

2.1 Taxonomy Expansion

Expanding existing incomplete taxonomies with new entities has been studied from several perspectives. Early efforts to extend a taxonomy are by detecting the hypernym relation of a ⟨query, anchor⟩ pair. They exploit the semantic relatedness between the query and anchor concepts, either by lexical patterns [12, 35] or distributional word representations [3, 9, 26]. However, these approaches usually fail to sufficiently explore the taxonomic hierarchies that encode structural semantics and knowledge. Recent works attempt to capture these hierarchies with the help of various structural summaries. A commonly used structural summary is *path*, a list of nodes connected by taxonomic edges. One state-of-the-art of using path is [42]. They first sample a set of top-down paths from the taxonomy. When predicting the true parent for a query, beyond just a candidate anchor, the classifier also has access to the semantic features of its structural contexts along the paths. Paths in taxonomy are further enhanced by a dynamic margin loss that compares the similarity between two paths in [18], and by language models that

[1] We note there are studies on asymmetric geodesic distance in certain spaces [24, 25], but most current non-Euclidean embeddings are in common spaces, such as hyperbolic space. Thus, we still focus on symmetric geodesic distances in common spaces.

formalize a taxonomy path as a pseudo linguistic sentence in [13]. Because an entity could have multiple parents or children, a set of sampled paths may not cover all the surroundings of an entity node in taxonomy. Therefore, [33] uses *local ego-graph*, which contains an entity with all its parents and children, to capture the local structures. They use graph neural networks[14, 36] to encode the local ego graph to boost the representation of the central entity. [39] extends the local ego-graphs to the root node and forms sub-trees, preserving more structural contexts. Recent works have begun to view a taxonomy from richer perspectives, including capturing heterogeneous semantics and relations [21, 40], representing taxonomy in non-euclidean spaces [2, 10, 20, 27], examining candidate parents and candidate children simultaneously for a query [44], and generating new concepts to fulfill the taxonomy [43].

However, almost all of these works represent entity nodes as high-dimensional vectors (i.e., points), which are only able to measure the *symmetrical* similarity (i.e., distance) between the two entities. Yet the hierarchies in taxonomy are inherently *asymmetrical*, such as the child-parent relation. Vector embeddings are not sufficient to differentiate the parent and child nodes in a pair, which limits their abilities to represent and expand a taxonomy. Our study instead learns box embeddings (i.e., high-dimensional rectangles) for entities, which naturally represent the asymmetrical hierarchical relations and is more appropriate for taxonomy expansion.

2.2 Representation Learning with Box

Different from the vector based embedding approaches, box embeddings represent objects or entities using geometric regions. It offers a more natural and intuitive way to model asymmetrical relations, such as hierarchies [15, 30] and transitive closure in directed graphs [15]. Box embeddings are initially established from the probabilistic perspective in [38], in which the box embeddings are learned by optimizing the conditional probability of two entities that form a hypernym. Despite the progress, optimizing the conditional probability upon the exact box edges has been shown to easily lead to training failure. [16] discloses the reason is that disjointed box pairs are difficult to optimize due to the lack of gradients. Therefore, the exact "hard" box edges are changed to "soft" by representing them with Gaussian density functions in [16] and Gumbel distributions in [7]. These "soft" boxes offer gradients for all training samples, enabling easier training, although are not intuitively interpretable to humans. Beyond the probabilistic view, box embeddings are also learned by capturing the geometric properties. [32] defines a geodesic distance between a vector and a box, and optimizes the box generator with a loss function designed upon this distance. Different from these works, we propose to learn the box embeddings from a joint view of geometry and probability. The core advantage of this joint view is that it provides an alternative approach to address the gradients missing problem, but still preserves the interpretability of exact "hard" boxes. We also show the joint view outperforms any single one empirically in Sec. 5.4.

Box embeddings have a wide range of applications. [16] detects hypernym upon the entailment of boxes; [4] uses the intersection of boxes to measure the uncertainty in knowledge graphs; [28] models both entity mentions and types as boxes to allow probabilistic fine-grained entity typing. Box embeddings have also shown success in word representation [6], knowledge base completion [1] and ranking [23]. In terms of application, to the best of our knowledge, our work is a very early attempt to expand taxonomy with geometrically-inspired embeddings. Different from many other tasks, one challenge of taxonomy expansion is to learn box embeddings for *new* entities that are compatible with existing taxonomy.

3 PRELIMINARY

3.1 Box Embeddings

Definition 3.1. (Box Embeddings [7, 16, 38]) A box embedding is a pair of vector embeddings that form a valid axes-aligned hyperrectangle in d-dimensional space.

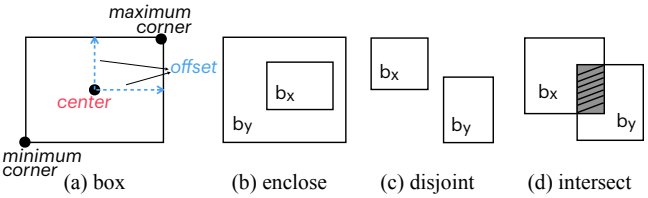

Figure 2: Illustration of box embeddings. (a) Characterization of a box. Note the offset is a 2-Dimensional vector in this example. (b) Enclosure relation: one box is fully contained within the other. (c) Disjoint relation: one box is fully outside the other. (d) Intersect relation: two boxes share overlap.

Box Characterization. A box (hyperrectangle) can be described by two vectors (points). Following [32], we use the center point and a positive offset vector to represent a box (Fig. 2 (a)). Denote by $b = (c, o)$ a d-dimensional box, where $c \in \mathbb{R}^d$ is the center and $o \in \mathbb{R}^d$ is the offset that is positive at all coordinates, we can then derive the minimum corner point as $l = c - o$ and maximum corner point as $r = c + o$, where $l \in \mathbb{R}^d$ and $r \in \mathbb{R}^d$. Note that a minimum and maximum corners pair can also define a box [28].

Volume. The volume of a box is the product of its segment in each dimension, formalized as $Vol(b) = \prod_{i=1}^{d}(r^i - l^i)$, where i is the indicator of dimension. If the entire box space's volume is 1, then the volume of a box can be modeled as its marginal probability.

Pairwise Relations. We elaborate on three pairwise relations between two boxes. Denote by $\langle b_x, b_y \rangle$ a pair of boxes, we have:

- *Enclose* (Fig. 2 (b)): a box b_x is completely contained inside the other box b_y, denoted by, $b_x \cap b_y = b_x$.
- *Disjoint* (Fig. 2 (c)): a box b_x is completely outside the other box b_y, denoted by $b_x \cap b_y = \emptyset$.
- *Intersection* (Fig. 2 (d)): a box b_x shares some overlap with the other box b_y, denoted by $b_x \cap b_y \neq \emptyset$.

The $\cap$ operator will return the intersection box b_z of the two boxes $\langle b_x, b_y \rangle$, denoted by $b_z = b_x \cap b_y$. Note that the $\cap$ operator is performed on the minimum and maximum corners, i.e., $\cap : l_z = max(l_x, l_y),\ r_z = min(r_x, r_y)$. The $\cap$ operator also enables the calculation of conditional probability between two boxes. Formally, $P(b_y|b_x) = \frac{P(b_x, b_y)}{P(b_x)} = \frac{Vol(b_x \cap b_y)}{Vol(b_x)}$.

3.2 Taxonomy Expansion

Definition 3.2. (Taxonomy [33, 42]) A taxonomy $\mathcal{T} = (\mathcal{E}, \mathcal{H})$ is a tree structure, where each node $e \in \mathcal{E}$ is an conceptual entity, and

each edge $h \in \mathcal{H}$ represents the "is-a" relation between the two nodes connected by it.

Taxonomy is usually incomplete, and new entities are constantly created, therefore, how to attach the new entities to an existing taxonomy is essential and the focus of this study. We formally define the taxonomy expansion problem as:

Definition 3.3. (Taxonomy Expansion [33, 42]) Given an existing taxonomy $\mathcal{T}^0 = (\mathcal{E}^0, \mathcal{H}^0)$ and a new conceptual entity set $\mathcal{N}$, taxonomy expansion aims to create a new taxonomy $\mathcal{T}' = (\mathcal{E}', \mathcal{H}')$, where $\mathcal{E}' = \mathcal{E}^0 \cup \mathcal{N}$ and $\mathcal{H}' = \mathcal{H}^0 \cup \mathcal{R}$. $\mathcal{R}$ is the set of newly created edges between $\mathcal{E}^0$ and $\mathcal{N}$.

4 PROPOSED METHOD: BOXTAXO

4.1 Overview

The core idea of BoxTaxo is to learn box embeddings for entities in taxonomy. Box embeddings are more natural and powerful in representing the asymmetrical taxonomic hierarchies, and thus can more accurately expand the taxonomy with new entities. As illustrated in Fig. 3, BoxTaxo first projects entities to box embeddings from natural language via the pre-trained language model (Sec. 4.2). During training, BoxTaxo optimizes the boxes from both the geometric view and the probabilistic view to precisely represent the hierarchies (Sec. 4.3). During inference, BoxTaxo encodes new query entities to boxes and finds the appropriate anchors in a probabilistic manner (Sec. 4.4).

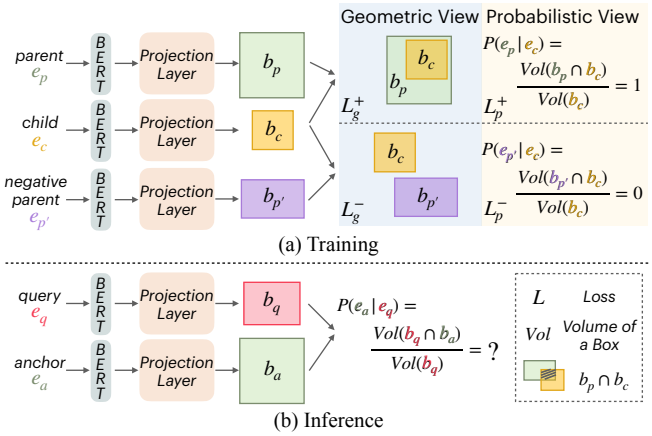

Figure 3: The overview of BoxTaxo. The entities in taxonomy are first projected to boxes based on Bert. (a) Training: the box embeddings are optimized from a joint view of geometry and probability, in order to accurately represent the taxonomic hierarchies. (b) Inference: check whether a query's box is enclosed by the candidate anchor's box in a probabilistic way. Note that the boxes shown in this figure are 2D, but they can be in higher dimension spaces, i.e., hyperrectangles.

4.2 Box Projection

The nodes in taxonomy are conceptual entities. To represent each entity as a box in the latent space, BoxTaxo uses a two-stage projection process. In specific, an entity is first encoded as numeric embeddings from natural language via an *entity encoder*, and then converted into a rectangular lattice by a *box projector*. We now introduce these two operators in detail.

Entity Encoder. The pre-trained language models (PTMs) have shown promising achievements in many natural language tasks [31]. Encouraged by their success, we use PTMs to encode the entities into embeddings. Without loss of generality, we use Bert [8] as the entity encoder in this paper. Formally, for the i-th entity e_i, Bert converts it into k-dimensional representation $n_i \in \mathbb{R}^k$:

$$n_i = \text{Bert}(e_i). \tag{1}$$

The entities in taxonomy are usually curated and thus could have definition sentences. Therefore, for such a definition sentence s_i, we concatenate it with its entity e_i and build the input of Bert as "`[CLS]`e_i, s_i `[SEP]`". We then use the output embeddings of "`[CLS]`" in the final Bert layer as the representation n_i of entity e_i. The representation n_i encodes the contextual semantics of the entity. Please note other pre-trained language models, such as Roberta [17] and ELECTRA [5], are flexible to be replaced as the encoder.

Box Projector. We then project the entity representation n_i into box embeddings. A box can be defined by two points (i.e., vectors). We therefore use the center point $c_i \in \mathbb{R}^d$ and the offset vector $o_i \in \mathbb{R}^d$ to represent a box $b_i \in \mathbb{R}^d$, i.e., $b_i = (c_i, o_i)$, where d is the dimension of box embeddings. Note that c_i and o_i are just two vector embeddings. To represent an entity e_i as box embeddings b_i, we project the entity representation n_i into the center c_i and offset o_i, separately. Specifically, because this projection is only a dimension transformation between two embeddings, we simply use two multilayer perceptrons (MLPs) as the projectors, formalized as:

$$c_i = \text{MLP}_\text{c}(n_i), \quad o_i = \text{MLP}_\text{o}(n_i), \tag{2}$$

where MLP_c and MLP_o are the projection layers for center c_i and offset o_i, respectively. To ensure the learned box b_i is a valid rectangle, we further apply an exponential operator to the offset o_i, so that every dimension of o_i is guaranteed to be larger than 0.

4.3 Box Training

We now seek to optimize the box embeddings such that they can accurately represent the taxonomic hierarchies, i.e., the child-parent relations. Because each ⟨child, parent⟩ pair in the taxonomy is a natural "label", we propose to fine-tune the entity encoder and box projector in a self-supervised manner. Specifically, we utilize all the immediate ⟨child, parent⟩ pairs in the taxonomy as positive samples. Negative samples have been demonstrated to be crucial in optimizing box embeddings [15]. Therefore, for each child node in such a pair, we collect its "siblings", "uncles" and "cousins" as the negative samples against the child-parent relations. Compared to vectors, box embeddings are more powerful in representing child-parent relations. We show how boxes achieve this advantage from two views: the *geometric* view and the *probabilistic* view. Accordingly, we design two training objective functions, the geometric loss and the probabilistic loss, to jointly optimize the box embeddings.

Geometric View. We first show how the child-parent relation can be represented with box embeddings in geometric language. A box with d-dimensional center and offset vectors is a d-dimensional hyperrectangle in Euclidean space. A ⟨child, parent⟩ pair can be semantically interpreted as "child *is-a* parent" or "child *is-one-of*

parent" [12]. Therefore, we let the child hyperrectangle be fully *enclosed* by the parent hyperrectangle, indicating the child entity is one kind of parent. Formally, for a d-dimensional child box $b_c = (c_c, o_c)$, since o_c is regularized to be positive, we denote by $l_c = c_c - o_c$ and $r_c = c_c + o_c$ the minimum and maximum corner points of the hyperrectangle, respectively. Similarity, for parent box $b_p = (c_p, o_p)$, denote by $l_p = c_p - o_p$ and $r_p = c_p + o_p$ the minimum and maximum corner points. Then the "enclose" relation has:

$$l_c^i \geq l_p^i, \quad r_c^i \leq r_p^i, \quad \forall i \in \{1, 2, ..., d\}, \tag{3}$$

where i denotes the i-th dimension of the embeddings. We derive a loss function L_g^+ to ensure boxes satisfy this geometric enclose relation (Eq. (3)) for pair $\langle e_c, e_p \rangle$, formalized as:

$$L_g^+ = \frac{1}{d}\Big[\sum_{i=1}^{d} max(0, l_p^i - l_c^i + \delta) + \sum_{i=1}^{d} max(0, r_c^i - r_p^i + \delta)\Big], \tag{4}$$

where δ is a hyper-parameter across all d dimensions that controls the geometric margin between the child and parent boxes.

Oppositely, for a negative pair $\langle$child, parent$'\rangle$, denoted by $\langle e_c, e_{p'} \rangle$, the child hyperrectangle should be *disjoint* with the negative parent hyperrectangle. We implement this "disjoint" relationship by enforcing the intersection between the child box and the negative parent box to be empty. Formally, for such a box pair$\langle b_c, b_{p'} \rangle$, their intersection $b_z = b_c \cap b_{p'}$ is formalized as:

$$l_z = max(l_c, l_{p'}), \quad r_z = min(r_c, r_{p'}). \tag{5}$$

An empty intersection, i.e., $b_z = \emptyset$, essentially means every dimension of the intersection b_z is less than or equal to 0. Based on this property, we derive a loss function L_g^- to minimize the offset o_z of the intersection, formalized as:

$$L_g^- = \frac{1}{d}\sum_{i=1}^{d}(o_z^i - \epsilon)^2, \tag{6}$$

where ϵ is a hyper-parameter to adjust the margin of intersection. If $\epsilon > 0$, we allow some intersection between two boxes, and when $\epsilon \leq 0$, we force the two boxes to be separated. Note that the offset can be derived by $o_z = (r_z - l_z)/2$.

Probabilistic View. We now introduce how the child-parent relation is represented by box embeddings from a probabilistic perspective. We first define taxonomic probability:

Definition 4.1. (Taxonomic Probability) Taxonomic probability $P(e_y|e_x)$ is the likelihood of event "from a given entity e_x, another entity e_y can be reached along a given 1-length edge" occurring.

For a $\langle$child, parent$\rangle$ pair $\langle e_c, e_p \rangle$ in taxonomy, the taxonomic probability $P(e_p|e_c) = 1$, because given a child, its exact parent can always be retrieved along the edge connecting them. If a child has multiple parents, we define the taxonomic probability as 1 for all parents. Similarly, for a negative pair $\langle$child, parent$'\rangle$, denoted by $\langle e_c, e_{p'} \rangle$, since the negative parent can not be directly reached given the child, the taxonomic probability $P(e_{p'}|e_c) = 0$. Desired box embeddings should satisfy these conditions of taxonomic probability for both positive and negative pairs, so that they can accurately represent the child-parent hierarchies in taxonomy.

Similar to using diagrams of sets to describe probabilities (i.e., Venn diagram [37]), box embeddings provide a natural graphical way to calculate the taxonomic probability. Following [16, 28, 38], we use the volume of the intersection between child box and parent box, divided by the volume of child box, to represent the taxonomic probability $P(e_p|e_c)$, formalized as:

$$P(e_p|e_c) = \frac{Vol(b_p \cap b_c)}{Vol(b_c)}, \tag{7}$$

where the $Vol(\cdot)$ is the volume of a box. On this basis, we propose a probability loss function for each positive child-parent pair $\langle e_c, e_p \rangle$, denoted by L_p^+, which is formalized as:

$$L_p^+ = (P(e_p|e_c) - 1)^2, \tag{8}$$

and also a probability loss function for each negative pair $\langle e_c, e_{p'} \rangle$, denoted by L_p^-, which is formalized as:

$$L_p^- = (P(e_{p'}|e_c) - 0)^2. \tag{9}$$

Box Regularization. In both geometric and probabilistic views, we design loss functions that minimize the intersection of two negative box embeddings, i.e., the negative geometric loss L_g^- and the negative probabilistic loss L_p^-. Actually, if a box is near zero in all its embedding dimensions, or its volume is close to zero, these two losses are also able to be minimized. In this case, however, the learned box embeddings are meaningless and can hardly represent the taxonomic hierarchies. To avoid this "cheating" during training, we regularize that box embeddings can not be too small in all dimensions. For box embeddings b_e of entity e, we implement this constraint by regularizing the offset o_e with regularization loss L_r:

$$L_r = \frac{1}{d}\sum_{i=1}^{d} min(0, o_e^i - \phi)^2, \quad \forall i \in \{1, 2, ..., d\}, \tag{10}$$

where ϕ controls the minimum length of boxes in each dimension.

Joint Loss. Finally, we combine the geometric losses, the probabilistic losses and the regularization loss to jointly train the model. Formally, the final loss function is:

$$L = \alpha(L_g^+ + L_g^-) + \beta(L_p^+ + L_p^-) + \gamma L_r, \tag{11}$$

where the α, β and γ are hyper-parameters to control the contributions of each single loss function.

Benefit of Joint View. Most box embeddings studies build the training objective from only the probabilistic view, where the core is to compute the volume of two boxes' intersection [4, 7, 16, 28, 38]. Following [16], denote by $b_z = b_c \cap b_p$ the intersection of a $\langle$child, parent$\rangle$ pair's boxes, then we rewrite its volume of as:

$$Vol(b_z) = \prod_{i=1}^{d} max(0, (min(r_c^i, r_p^i) - max(l_c^i, r_c^i))). \tag{12}$$

Eq. (12) is a hinge loss, where the (sub)-gradient is 0 when two boxes are disjoint, i.e. $min(r_c^i, r_p^i) - max(l_c^i, r_c^i) \leq 0$. This leads to a serious issue, namely that if two boxes of a true $\langle$child, parent$\rangle$ pair are accidentally disjoint during training, these incorrect boxes will never be optimized if only the probabilistic view is included in the loss function. [7, 16] propose to represent the edges of a box as probabilistic density functions, so that the gradients always exist even the two boxes are disjoint. However, making boxes "soft" will lose their natural and intuitive interpretability to humans. Our joint view of geometry and probability is an alternative approach to this "zero gradients" issue but still preserves the interpretability of "hard" boxes. Specifically, for child-parent pairs that are falsely disjoint,

the geometric loss Eq. (4) will provide gradients for optimization, because at least one of $l_p^i - l_c^i + \delta$ and $r_c^i - r_p^i + \delta$ is greater than 0. Note Eq. (4) is for positive pairs. For negative pairs, the two boxes are expected to be disjoint, so there is no gradient issue.

In conclusion, the core benefit of the joint view of geometry and probability is to ensure that gradients are always available during optimization without losing the natural interpretability of boxes.

4.4 Inference with Box

During inference, our goal is to find an appropriate parent entity, i.e., an anchor, from the taxonomy for a given new query. In contrast to vector embeddings that measure the distance between two points, box embeddings are more intuitive and natural to determine whether a candidate parent is suitable: by checking to what extent the box of anchor encloses the box of the query. We implement this idea in a probabilistic way as shown in Fig. 3 (b). Specifically, for query e_q, we first project it into a box b_q and then compare it with each candidate anchor e_a's box b_a[2]. Formally, we rank the candidates by their taxonomic probabilities $P(e_a|e_q)$. A higher $P(e_a|e_q)$ indicates that anchor e_a is more likely to be an appropriate parent for the query e_q. In some cases, the taxonomic probability values of many anchors could be the same. For example, if the query box is enclosed by the box of a leaf anchor node, then it is also enclosed by all ancestors (until the root) of this leaf anchor, i.e., $P(e_a|e_q) = 1$ for the leaf anchor and all its ancestors. In this case, we return this leaf anchor as the predicted parent, since it is the finest-grained and more precisely describes the query. We note this leaf anchor should have the smallest volume because it is enclosed by all ancestors. Therefore, for candidate anchors with the same taxonomic probability, we perform a second ranking according to the volume of their boxes, so that finer-grained anchors can be placed higher.

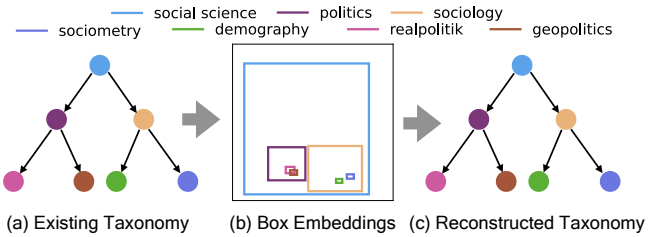

Figure 4: Case study of reconstructing the existing taxonomy from the learned box embeddings. A sub-tree rooted in "social science" from the Science dataset is shown.

5 EXPERIMENTS

5.1 Setup

Datasets. Following [42], to evaluate how BoxTaxo works in taxonomy expansion, we use two public datasets from SemEval-16 taxonomy construction tasks. The taxonomy entities are scientific concepts in the environment field and in general science, respectively. The human-curated hierarchies represent the category of each entity. We use the same data splitting protocol as [42], i.e., 20% of the leaf nodes are randomly sampled as the test set, while the remaining are in the training set. Definitions for each entity are also provided in both datasets. We simply combine entity names and their definitions as input to the model.

Experimental Settings. We compare BoxTaxo with five vector based baselines for taxonomy expansion. We introduce the details of baselines in Appendix. A.1. We use three metrics, Accuracy (ACC), Mean reciprocal rank (MRR) and Wu & Palmer similarity (Wu&P) [41], to measure the performance of BoxTaxo compared to baselines. We present the details and formalization of all metrics in Appendix. A.2. We also list the key parameters used in BoxTaxo for reproducibility in Appendix. A.3.

5.2 Can the Learned Box Embeddings Reconstruct the Existing Taxonomy?

One motivation for this paper is that box embeddings are better at representing the hierarchies in taxonomy. To verify this, we reconstruct the existing (training) taxonomy, i.e., predict the child-parent relations, from the learned box embeddings. Overall, the hierarchies are well preserved by box embeddings: we achieve 82.2% on Environment dataset and 60.9% on Science dataset in terms of reconstruction accuracy. To intuitively show how the learned boxes capture the taxonomic hierarchies, we sample a sub-tree rooted in "social science" from the Science dataset, and show the learned boxes in Fig. 4. We notice the two branches are fully separated and every child is enclosed by their parents, indicating the hierarchies are fully captured. With such hierarchy-aware boxes, we then study how taxonomy expansion benefits from them in the following.

Table 1: Results of BoxTaxo on taxonomy expansion compared to vector based methods. We use the same experimental setting as [42] and the baseline results are from [42]. We report the averages of ten runs of BoxTaxo. The best results are in boldface, and the second-best results are underlined. The "N/A" indicates that MRR is not applicable to TAXI.

Dataset	Environment			Science		
Metric	ACC	MRR	Wu&P	ACC	MRR	Wu&P
TAXI	16.7	N/A	44.7	13.0	N/A	32.9
HypeNet	16.7	23.7	55.8	15.4	22.6	50.7
Bert+MLP	11.1	21.5	47.9	11.5	15.7	43.6
TaxoExpan	11.1	32.3	54.8	27.8	44.8	57.6
STEAM	36.1	46.9	69.6	**36.5**	**48.3**	**68.2**
BoxTaxo	**38.1**	**47.1**	**75.4**	31.8	45.3	64.7

5.3 Are Box Embeddings Better than Vector Embeddings for Taxonomy Expansion?

We compare BoxTaxo with vector based embeddings baselines for taxonomy expansion and report the results in Table. 1. We include two lines of baselines: 1) Because BoxTaxo only models the simple ⟨child, parent⟩ pairs during training, we first compare BoxTaxo with vector based counterparts that also focus on such pairs, i.e., TAXI, HypeNet and Bert+MLP. BoxTaxo outperforms them with significant gains, indicating the effectiveness of box embeddings against vectors for taxonomy expansion. 2) We also

[2]Here we use all entities in the existing taxonomy as the candidate anchor set.

compare BoxTaxo with vector based baselines that use advanced structural summaries, including local graphs (TaxoExpan) and paths (STEAM). Despite not explicitly modeling such structural signals, BoxTaxo still achieves a clear improvement over TaxoExpan and shows comparable results with STEAM. We are encouraged by these results as it shows the potential to facilitate box embeddings with advanced structures to further boost taxonomy expansion. Interestingly, we notice that compared to baselines, BoxTaxo shows better accuracy on the Environment dataset than on the Science dataset. Actually, the edge-to-node ratio is 1 on the Environment dataset and 1.05 on the Science dataset, indicating that some nodes in the Science dataset have multiple parents. We find these "multi-parent" nodes may bring all nodes that are associated with their parents closer in the box embeddings space, causing incorrect predictions. We did not identify an adequate solution to address this issue and thus leave it as a topic for future investigation.

5.4 Ablation Study: Does BoxTaxo Benefit from the Joint Loss of Geometry and Probability?

BoxTaxo optimizes box embeddings from both geometric and probabilistic views. To examine the benefits of this joint objective, we perform an ablation study that learns box embeddings only with each single view. The results are reported in Table. 2. We note that learning boxes with the joint view shows clear superiority in taxonomy expansion. These results echo our discussion in Sec. 4.3 that the joint view objective function can ensure that gradients for optimization are always available, thereby enhancing box learning.

Table 2: Ablation study of the joint view optimization, i.e., Eq. (11). "No Geo" means learning box embeddings only with the probabilistic loss, while "No Prob" is only with the geometric loss. "Joint" stands for optimizing with both losses. We fix the box dimensions as 12 for both datasets.

Dataset	Environment			Science		
Metric	ACC	MRR	Wu&P	ACC	MRR	Wu&P
No Geo	12.8	30.6	58.4	7.7	21.5	50.4
No Prob	15.8	25.8	59.2	30.1	41.1	64.4
Joint	**35.3**	**44.8**	**74.2**	**31.8**	**45.3**	**64.7**

5.5 Can Box Embeddings Capture Implicit Hierarchical Relations in Taxonomy?

Beyond the basic ⟨child, parent⟩ pairs, a taxonomy also includes implicit hierarchical structures. We study two common implicit relations: *transitivity* and *overlap*. Transitivity refers to "the parent of a child's parent is also this child's parent (ancestor)", e.g., in "natural science→physics→mechanics", we know that "natural science→mechanics" also holds. Overlap refers to "two parents share common children", e.g., "morphology" is an entity both in "biology" and "grammar" fields. We sample a sub-structure for each of these two implicit relations from Sciecne dataset and visualize their corresponding learned boxes in Fig. 5. We note that for transitivity, a child's box is always enclosed by the boxes of all its ancestors. For overlap, the common child's box is enclosed in the intersection

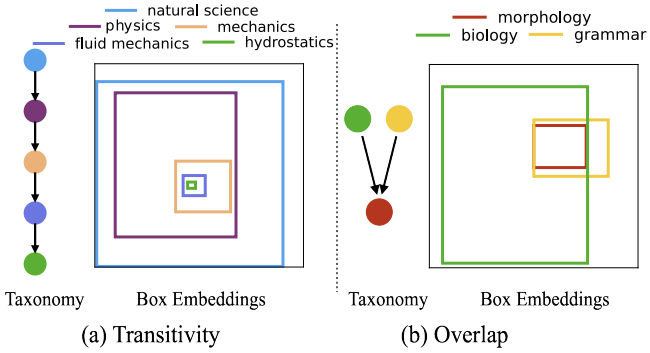

Figure 5: Two implicit hierarchical relations in taxonomy (sampled from Science dataset) and how the learned box embeddings preserve them. (a) Transitivity, (b) Overlap.

area of its two parents. Both indicate the box embeddings learned by BoxTaxo well preserve the implicit hierarchical relations. The capability of representing such implicit relations goes along with the advantage of boxes over vectors in taxonomy representation.

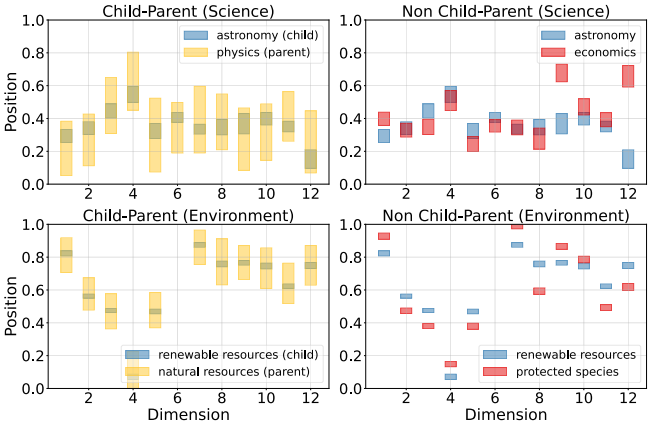

Figure 6: Comparison of two boxes in each of the 12 dimensions. A (blue, yellow) is a child-parent pair in taxonomy, while a (blue, red) is not. Overlap means two boxes intersect in a dimension. Upper row: Science; lower row: Environment.

5.6 How BoxTaxo Works in High-Dimensional Space?

Child-Parent Learning in High-Dimensional Space. We then analyze whether the learned boxes can still preserve the taxonomic hierarchies in higher dimensions. A box is a hyperrectangle in high-dimensional space, where its edge in each dimension is a line segment. We plot the edges of a box pair in each dimension to study their overlap in Fig. 6. Specifically, we sample a positive and a negative child-parent pair for both datasets. We observe that, for the positive pairs of the two datasets, the edge of the child box is enclosed by the edge of the parent box in each dimension, indicating the child-parent relations are well captured. While for the negative pairs, we note the edges of the two boxes are separated in Environment dataset, but still overlap in some dimensions in

Science dataset. This overlap indicates that the negative pairs are not completely disjoint as we expect, which might explain why BoxTaxo performs worse in Science than in Environment dataset.

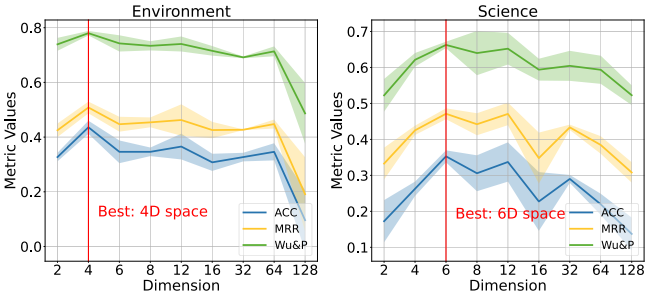

Figure 7: Three metrics ACC, MRR and Wu&P with nine different dimensions of box space on both datasets. Solid curves indicate mean, and ribbons show the standard derivation. The red vertical lines indicate the dimensional space where BoxTaxo works best. Left: Environment; Right: Science.

Taxonomy Expansion in High-Dimensional Spaces. To understand how boxes learned in different dimensions affect taxonomy expansion in BoxTaxo, we vary the box dimensions and report the corresponding taxonomy expansion metrics in Fig. 7. For both datasets, as the dimension increases, the metrics first get better and then slowly decrease. We speculate that BoxTaxo needs enough space to hold the entity, so the dimensions can not be too small. However, an excessively large dimension can lead to optimization difficulties and thus downgrade taxonomy expansion.

5.7 Case Study: When BoxTaxo Makes Mistakes?

To intuitively understand the predictions made by BoxTaxo and analyze when BoxTaxo fails, we sample two correct and two wrong predictions from the Science dataset in Table. 3. For correct cases, as we expect, the child's box is tightly enclosed by its true parent. For query "neuroanatomy", we speculate that our top-2 predictions, "neuroscience" and "neurobiology", lean more towards the semantics of the "neuro" part. However, we believe they are also reasonable anchors, though not listed in the dataset. This suggests the potential of BoxTaxo to discover multiple parents for cross-category entities. However, for query "marine archeology", we speculate it shares some semantic similarity with the wrongly predicted parent "zoology". Thus they are encoded closely in the box space. Yet, we note the true anchor "archeology" is the second prediction, we think BoxTaxo also can be used in applications where the goal is to retrieve several anchor candidates, e.g., the search engines.

5.8 How Hypermeters Affect BoxTaxo?

Finally, we study the impacts of hyperparameter δ, ϵ and ϕ on taxonomy expansion. We vary each hyperparameter with nine different values and report their corresponding metrics in Fig. 8. We notice BoxTaxo is sensitive to the margins of geometric loss functions δ and ϵ, i.e., they can not be too large. But a larger ϕ generally increases the performance of BoxTaxo, especially on the Science dataset, demonstrating the necessity of volume regularization.

Table 3: Case studies of BoxTaxo on Science dataset. Entities and their boxes are in a one-to-one correspondence by color.

Query and Anchor	Top3 Prediction	Box (2-Dimensional)
Q: linear algebra A: algebra	algebra, ✓ anthropology, pure mathematics	
Q: celestial mechanics A: astronomy	astronomy, ✓ physics, medicine	
Q: neuroanatomy A: anatomy	neuroscience, ✗ neurobiology, mechanics	
Q: marine archeology A: archeology	zoology, ✗ archeology, geology	

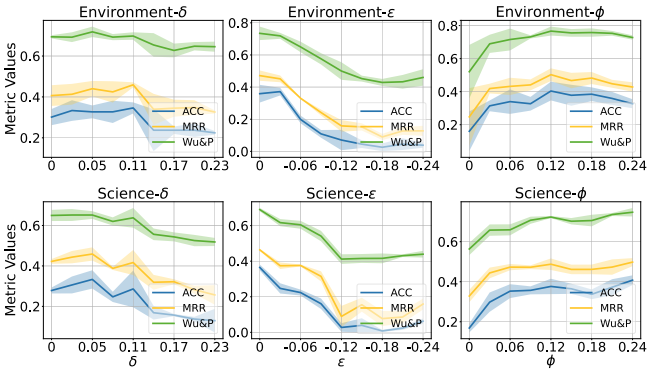

Figure 8: Three metrics ACC, MRR and Wu&P with different values of hyperparameter δ, ϵ and ϕ on both datasets. Solid curves indicate mean, and ribbons show the standard derivation. Upper row: Environment; Lower row: Science.

6 CONCLUSION

In this paper, we propose BoxTaxo, a novel self-supervised model for expanding an existing taxonomy with new entities. Since the taxonomic hierarchies are naturally asymmetrical relations, BoxTaxo learns box embeddings, instead of the traditional vector embeddings, to represent and expand a taxonomy. We propose to optimize the box embeddings from both geometric and probabilistic views to capture the taxonomic hierarchies. Extensive experiments show BoxTaxo is able to well preserve the hierarchical structures in taxonomy and outperforms the vector based baselines clearly. We also realize that incorporating lexical features, as demonstrated in [42], and structural signals such as paths and local graphs, has great potential to further enhance box embeddings learning and taxonomy expansion, which we leave to future works.

ACKNOWLEDGMENTS

The authors express their gratitude to Jie Tang, Qingyang Zhong, Jifan Yu, and Kewei Cheng for their insightful discussions. This work was partially supported by NSF 2211557, NSF 1937599, NSF 2119643, NASA, SRC, Okawa Foundation Grant, Amazon Research Awards, Cisco research grant, Picsart Gifts, and Snapchat Gifts.

REFERENCES

[1] Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems* 33 (2020), 9649–9661.

[2] Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019. Every Child Should Have Parents: A Taxonomy Refinement Algorithm Based on Hyperbolic Term Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4811–4817.

[3] Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional Inclusion Vector Embedding for Unsupervised Hypernymy Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, New Orleans, Louisiana, 485–495.

[4] Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew McCallum. 2021. Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online.

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[6] Shib Dasgupta, Michael Boratko, Siddhartha Mishra, Shriya Atmakuri, Dhruvesh Patel, Xiang Li, and Andrew McCallum. 2022. Word2Box: Capturing Set-Theoretic Semantics of Words using Box Embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[7] Shib Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. 2020. Improving local identifiability in probabilistic box embeddings. *Advances in Neural Information Processing Systems* 33 (2020).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Minneapolis, Minnesota.

[9] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 1199–1209.

[10] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*. PMLR, 1646–1655.

[11] Rafael S Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A Musen, Csongor I Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. 2019. Use of OWL and semantic web technologies at pinterest. In *International Semantic Web Conference*. Springer, 418–435.

[12] Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

[13] Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. 2022. TaxoEnrich: Self-Supervised Taxonomy Completion via Structure-Semantic Representations. In *Proceedings of the ACM Web Conference 2022*. 925–934.

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[15] Alyssa Lees, Chris Welty, Shubin Zhao, Jacek Korycki, and Sara Mc Carthy. 2020. Embedding semantic taxonomies. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1279–1291.

[16] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. 2018. Smoothing the geometry of probabilistic box embeddings. In *International Conference on Learning Representations*.

[17] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* (2019).

[18] Zichen Liu, Hongyuan Xu, Yanlong Wen, Ning Jiang, Haiying Wu, and Xiaojie Yuan. 2021. TEMP: Taxonomy Expansion with Dynamic Margin Loss through Taxonomy-Paths. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3854–3863.

[19] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[20] Mingyu Derek Ma, Muhao Chen, Te-Lin Wu, and Nanyun Peng. 2021. HyperExpan: Taxonomy Expansion with Hyperbolic Representation Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic.

[21] Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*. 2044–2054.

[22] Yuning Mao, Tong Zhao, Andrey Kan, Chenwei Zhang, Xin Luna Dong, Christos Faloutsos, and Jiawei Han. 2020. Octet: Online catalog taxonomy enrichment with self-supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2247–2257.

[23] Lang Mei, Jiaxin Mao, Gang Guo, and Ji-Rong Wen. 2022. Learning Probabilistic Box Embeddings for Effective and Efficient Ranking. In *Proceedings of the ACM Web Conference 2022*. 473–482.

[24] Andrea CG Mennucci. 2013. On asymmetric distances. *Analysis and Geometry in Metric Spaces* 1, 2013 (2013), 200–231.

[25] Andrea CG Mennucci. 2014. Geodesics in asymmetric metric spaces. *Analysis and Geometry in Metric Spaces* 2, 1 (2014).

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[27] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems* 30 (2017).

[28] Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling Fine-Grained Entity Types with Box Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online.

[29] Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 1320–1327.

[30] Dhruvesh Patel and Shib Sankar. 2020. Representing joint hierarchies with box embeddings. *Automated Knowledge Base Construction* (2020).

[31] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* 63, 10 (2020), 1872–1897.

[32] Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[33] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*.

[34] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany.

[35] Rion Snow, Daniel Jurafsky, and Andrew Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in neural information processing systems* 17 (2004).

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[37] John Venn. 1880. I. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 10, 59 (1880), 1–18.

[38] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia.

[39] Suyuchen Wang, Ruihui Zhao, Xi Chen, Yefeng Zheng, and Bang Liu. 2021. Enquire one's parent and child before decision: Fully exploit hierarchical structure for self-supervised taxonomy expansion. In *Proceedings of the Web Conference 2021*. 3291–3304.

[40] Suyuchen Wang, Ruihui Zhao, Yefeng Zheng, and Bang Liu. 2022. QEN: Applicable Taxonomy Completion via Evaluating Full Taxonomic Relations. In *Proceedings of the ACM Web Conference 2022*. 1008–1017.

[41] Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033* (1994).

[42] Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1026–1035.

[43] Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2104–2113.

[44] Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. 2021. Taxonomy completion via triplet matching network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4662–4670.

A APPENDIX

A.1 Baselines

The scope of this work is to study whether box embeddings are more appropriate than vector embeddings for representing and expanding taxonomic hierarchies. Therefore, we mainly choose the vector based models. We also include two baselines that use advanced structural signals. We hope to learn whether BoxTaxo could achieve comparable results compared to structure-induced vectors, even without explicitly modeling those complex structures. The baselines we choose are:

- **TAXI** [29]. A hypernym detection based method. It constructs the taxonomy by determining hypernym between entity pairs on the basis of lexical patterns.
- **HypeNet** [34]. A vector based method using LSTM to encode dependency paths for entity pairs embeddings.
- **Bert+MLP** [42]. A vector based method using a pre-trained language model (Bert [8]) to generate entity embeddings.
- **TaxoExpan** [33]. A vector based method using graph neural networks (GNNs) to encode local ego-graphs in taxonomy to enhance entity representation.
- **STEAM** [42] A vector based method using paths sampled from taxonomy to improve anchor entity representation.

A.2 Evaluation Metrics

Now we introduce the metrics used in this paper for evaluating the performance of taxonomy expansion. For a new query, we can view the output of BoxTaxo and baselines as a ranking of all candidate entities in the existing taxonomy, according to their suitability to be anchors. Denote by a_i the true anchor for the i-th query, and denote by $\hat{a}_i$ the top-1 predicted anchor. We use three metrics to compare the performance of BoxTaxo with the baseline:

Accuracy (ACC): the precision of predicted anchors.

$$\text{ACC} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{a}_i = a_i).$$

Mean reciprocal rank (MRR): measures the position of the true anchor in the ranked output

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank(a_i)}.$$

Wu & Palmer similarity (Wu&P) [41]: a similarity measure that captures semantics in taxonomy

$$\text{Wu\&P} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\times \text{depth}(\text{LCA}(\hat{a}_i, a_i))}{\text{depth}(\hat{a}_i) + \text{depth}(a_i)}.$$

The $LCA(\cdot,\cdot)$ stands for the least common ancestor of two inputs, and the $depth(\cdot)$ is the depth of an anchor in taxonomy. N is the number of test queries for all metrics.

A.3 Reproducibility

The parameters of BoxTaxo are as follows. We use one hidden layer in box projection with dimension 64. The dimension of box embeddings is 4 for the Environment dataset and 12 for the Science dataset, respectively. For margins in loss functions, we set $\delta = 0.05$, $\epsilon = -0.03$, and $\phi = 0.03$. For the weights of each single loss, we set $\alpha = 1$, $\beta = 0.1$ and $\gamma = 1$. For the hyper-parameters in training, we use AdamW [19] to optimize our model with the learning rate 2e-5 for Bert and 1e-3 for the box projection MLP layers. The epsilon of AdamW is set to 1e-8. We train BoxTaxo on both datasets 100 epochs with batch size 100. The training hyperparameters are default values to ensure model convergence. All experiments are done on a server with Nvidia A100 GPUs.

A.4 Additional Experimental Results

In Sec. 5.3, we follow the data splitting protocol used in [42] to ensure a fair comparison. In this section, we also present taxonomy expansion results of BoxTaxo with a new dataset split: train/validation/test = 7/1/2. Under this new setting, BoxTaxo has ACC/MRR/Wu&P/=36.2/47.3/73.0 on Environment dataset, while on Science dataset, we have ACC/MRR/Wu&P/=29.2/40.1/62.2, which are still comparable to baselines. Note that compared to Table. 1, the results have decreased due to a reduced size of the training set.