Neural Processing Letters (2024) 56:102
https://doi.org/10.1007/s11063-024-11575-3

PEB-TAXO: Projecting Entities as Boxes for Taxonomy Expansion

Yuhang Zhang[1,2] · Jiwei Qin[1,2] · Chongren Feng[1,2]

Accepted: 15 February 2024 / Published online: 12 March 2024

Abstract
As domain knowledge evolves, new concepts (entities) continuously emerge, leading to a decrease in the coverage of existing taxonomies with hierarchical structures, thus necessitating the continual expansion of these taxonomies to include new concepts. Due to the relationships ("contain", "disjoint", and "intersect") between the boxes, which can effectively represent asymmetric hierarchies, box embeddings have been successfully applied in taxonomy expansion. However, existing models that use box embeddings for taxonomy expansion have the following shortcomings: (1) the size of the boxes is not restricted, and the model produces meaningless boxes; (2) the model does not fully utilize the geometric information of the boxes. To address the above shortcomings, this paper proposes a taxonomy expansion model based on projecting entities as boxes: PEB-TAXO. Firstly, PEB-TAXO employs modified L1 regularization to constrain the box sizes in all dimensions, pushing the box sizes towards the preset minimum, thereby avoiding the generation of meaningless boxes by the model. Secondly, the model utilizes a box inclusion inference method: it infers the relationship between two entities through the relationship between two boxes in geometric space, thus fully exploiting the geometric information of the boxes for more accurate inferences. Finally, we conducted extensive experiments on two public datasets and verified that PEB-TAXO greatly improves performance over mainstream taxonomy expansion methods.

Keywords Taxonomy · Taxonomy expansion · Box embeddings · Modified L1 regularization · Box inclusion

✉ Jiwei Qin
jwqin@xju.edu.cn

Yuhang Zhang
107552103782@stu.xju.edu.cn

Chongren Feng
107552103587@stu.xju.edu.cn

[1] School of Computer Science and Technology, Xinjiang University, Ürümqi 830046, China

[2] Xinjiang Key Laboratory of Signal Detection and Processing, Xinjiang University, Ürümqi 830046, China

1 Introduction

Taxonomy is an essential representation in domain knowledge, which organizes concepts into a hierarchical taxonomic structure and uses tree diagrams to represent the hierarchical relationships (also known as parent–child relationships) between concepts [1]. Due to the efficiency of taxonomies in organizing knowledge concepts, taxonomies have become an indispensable component in knowledge systems such as knowledge graphs [2] and are widely used in many downstream applications of knowledge systems such as recommender systems, Q&A systems, search systems, and more [3–6]. However, with the continuous development of domain knowledge and the rapid growth of new concepts, many existing taxonomies face the issue of reduced coverage. Existing taxonomies are primarily curated by domain experts [7], but updating and expanding these taxonomies require significant time, effort, and financial resources. Therefore, researchers have utilized new concepts to automatically extend existing taxonomies to address the reduced coverage problem, thus avoiding taxonomic obsolescence.

Researchers have proposed the task of Taxonomy Expansion (TE) to achieve automatic expansion of existing taxonomies. The goal of this task is to find the parent concepts ("anchors") [1, 2] for new concepts ("queries") [1, 2] within the existing taxonomy, to integrate the new concepts into the taxonomy. For example, as shown in Fig. 1, with increasing attention to environmental topics, many new concepts ("New Concepts" section in Fig. 1) emerge in the environmental domain. To properly incorporate these concepts into the existing taxonomy "Environmental Policy" ("Existing Taxonomy" section in Fig. 1), researchers need to find the parent concepts ("Exploitation of Resources", "Management of Resources", etc. in Fig. 1) for these new additions and establish parent–child relationships to achieve taxonomy expansion ("Expanded Taxonomy" section in Fig. 1).

Common approaches to taxonomy expansion focus on capturing the parent–child hierarchy between queries and taxonomy nodes. In earlier studies, researchers used the semantic relationship between two entities to learn the hierarchy. The semantics can be represented in terms of lexical patterns [8, 9] or more powerful distributional word embeddings [10–12]. In addition to semantic relations, recent researchers have begun to model the tree structure of taxonomies. They use structural summaries such as local graphs [1, 13, 14] and paths [2, 15, 16] as additional signals to enhance parent–child hierarchy learning and better capture the parent–child hierarchy between entities.

The core idea of the methods above is to learn vector embeddings for entities in the taxonomy and then infer the existence of parent–child relationships between them by computing their correlation in geometric space. However, the essence of vector embeddings is to transform the semantic information of entities in the taxonomy into vector representations, represented as points in the embedding space. Using vector embeddings to describe the correlation between entities is actually using the Euclidean distance to measure the similarity between two points, a measure with symmetry. But the parent–child hierarchy in taxonomy is asymmetric and closer to the form of inclusion. Therefore, the vector embedding-based approach cannot represent the taxonomy hierarchy well, but somewhat limits the implementation of the taxonomy expansion task.

To overcome the drawbacks of vector embedding, we use boxes to represent the entities in the taxonomy, i.e., box embedding [17–19]. Unlike the representation of a vector as a single point in geometric space, a box is represented as a hyper-rectangle with a geometric region, allowing them to depict complex and asymmetric relationships, such as "contains", "intersects", and "disjoint". The "Management of Resources" inside the "Expanded Taxonomy" in Fig. 1 and all its subordinate nodes are modeled as boxes, as shown in Fig. 2. We can observe

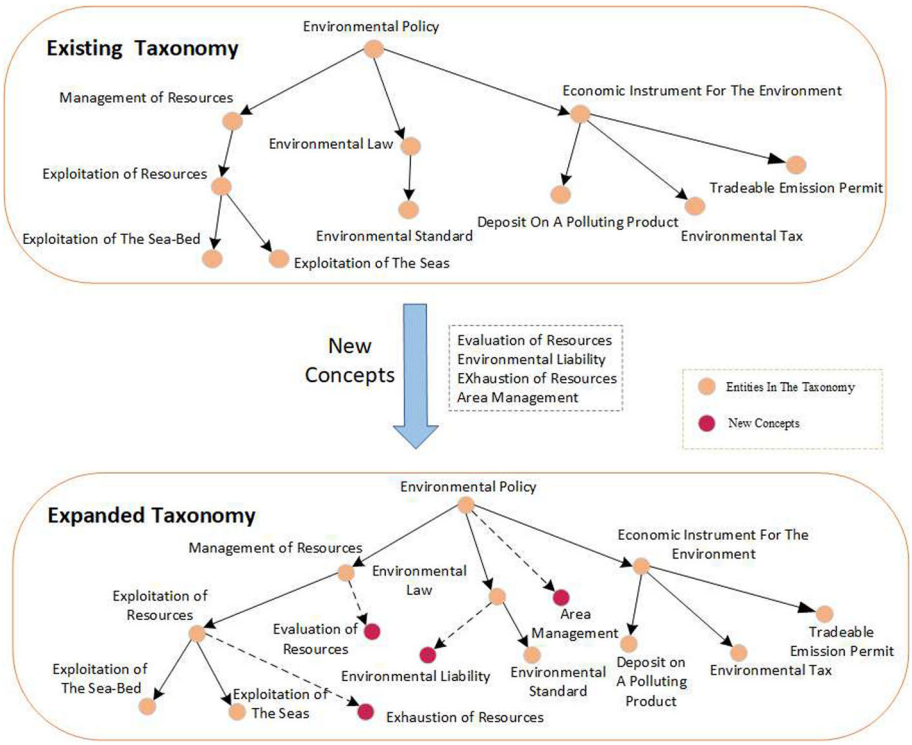


Fig. 1 Example of taxonomy expansion

that box embeddings clearly showcase the hierarchical structure of the taxonomy. Although box embeddings can naturally and intuitively represent the hierarchical structure of the taxonomy, there are still two main challenges in applying them to taxonomy expansion tasks: (1) many models that use box embeddings for taxonomy expansion continuously decrease the size of the boxes during the training phase to improve training efficiency, making them infinitely close to zero. However, this practice generates numerous meaningless boxes, resulting in a reduction in the robustness of the models using such embeddings; (2) mainstream box embedding methods in the inference stage of taxonomy expansion tasks often rely on probabilistic approaches [20], which do not fully utilize the geometric information of the boxes. As a result, they are susceptible to data distribution interference, leading to inaccurate inference results when noise exists in the training data. Therefore, further improvements are needed in box embedding methods to better leverage their advantages in taxonomy expansion tasks.

To address the above problem, we propose a self-supervised model for extending the taxonomy using box embeddings, called PEB-TAXO, which has two crucial components: box training and box inference. Specifically, when we perform box training, we add a modified L1 regularisation to the model, which pushes the box towards sparsity by penalizing the absolute value of each dimension of the box, so that the box size tends to the set minimum length greater than zero in each dimension, thus effectively controlling the size of the box to keep it in a suitable range, as a way of solving the problem of the model generating meaningless boxes. Then, In the inference stage, we calculate the coordinates of the corner points of

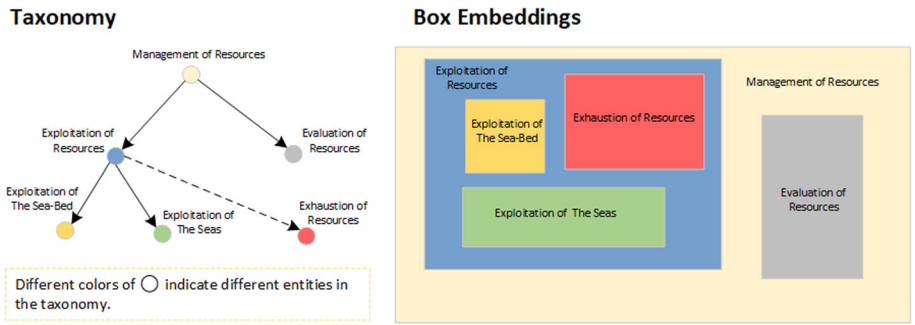


Fig. 2 Example of box embeddings

the child and parent boxes [21, 22]. By comparing the positions of these corner points, we determine whether the child box is entirely contained within the parent box. Based on the containment relationship, we generate corresponding containment indexes (calculated from the corner point positions, see Sect. 3.3), enabling us to make clear-cut judgments on the parent–child hierarchical structure using the geometric information of the boxes. Finally, we conduct extensive experiments on two real-world datasets and validate the effectiveness of our model. The results clearly show that our model performs significantly better than existing approaches. In summary, our main contributions to this work are as follows:

(i) During the box training process of PEB-TAXO, we introduce a modified L1 regularization to ensure the existence of the boxes and prevent their sizes from approaching zero indefinitely. This prevents the generation of meaningless boxes, thus enhancing the robustness of the model;
(ii) This paper proposes the box inclusion inference method to speculate whether two entities are parent–child hierarchies, in this way to make full use of the geometric information of the box, so that the model has richer information for speculation, which improves the accuracy of the model speculation;
(iii) Numerous experiments are conducted on two real datasets to verify the validity and stability of PEB-TAXO. And PEB-TAXO has a better performance compared with mainstream methods.

2 Related Work

2.1 Taxonomy Construction

Taxonomy construction aims to create a tree-structured taxonomy from scratch that contains a set of terms (such as concepts and entities) and integrates contextual relationships. The process can be further subdivided into two aspects. The first aspect focuses on topic-based taxonomies, where each node clusters several terms sharing the same topic [23, 24]. The second aspect addresses the construction of a term-based taxonomy, where each node represents a term itself [25]. Typically, the typical flow of this task is to first extract the "is-a" relations in the superordinate word detection model using pattern-based models [26, 27] or distributional models [28, 29]. Then, the mined hierarchical relationships are integrated and pruned into a directed acyclic graph or tree. In this way, the process of taxonomy construction is achieved.

2.2 Taxonomy Expansion

In terms of taxonomy expansion, several studies have explored it from different perspectives. Initial approaches focused on extending taxonomies by detecting the parent–child relationships between queries and anchor nodes, mainly relying on their semantic correlations, such as using lexical patterns [8, 9] or distributional word representations [10–12]. However, these approaches fail to fully explore the semantics of encoded structures and the categorization level of knowledge. Recent work has attempted to capture these hierarchies using different structural summaries, one typical summary being paths, which are lists of nodes connected by categorical edges. The paths state-of-the-art technique STEAM [2] exemplifies a set of top-down paths in a taxonomy. STEAM converts the taxonomy expansion task into a classification task on mini-paths. The model's classifier accesses the taxonomy hierarchy through paths when predicting the actual parent node of a query. It further employs three sub-models to process the taxonomy's distributional features, contextual features, and lexical-syntactic features, integrating them for enhanced performance. Another study used a local ego graph [1] to capture the local structure of an entity, which contains an entity with all its parents and children. The researchers used graphical neural networks to encode the local ego-graph to enhance the representation of the central entity. Apart from the abovementioned perspectives, recent research has started to approach taxonomy from various angles. For instance, ETF [30] trained a learning-to-rank framework using manually crafted structural and semantic features. It leveraged domain-specific functional knowledge to enrich the taxonomy, integrating domain knowledge and concepts to enhance its coverage and semantic relevance. Emaad Manzoor [24] and his colleagues utilized the Arborist tool to analyze implicit associations between entities, capturing relationships and semantic information that was not explicitly represented in the taxonomy. This approach expanded the taxonomy's hierarchical structure and semantic relationships. However, these methods represent entity nodes as high-dimensional vectors (i.e., points), which can only measure the symmetric similarity between two entities, contradicting the asymmetric parent–child hierarchy in taxonomy. Therefore, vector embedding-based methods cannot effectively represent the hierarchical structure in taxonomy, limiting their capability in representing and extending taxonomies. Instead, our study employs the projection of entities as boxes (i.e., high-dimensional rectangles), which naturally represent asymmetric hierarchical relationships and are more suitable for taxonomy expansion tasks.

2.3 Representation Learning with Box

Unlike vector-based embedding methods, box embedding uses geometric regions to represent objects or entities, providing a more natural and intuitive way of modeling asymmetric relationships. In the early research on box embeddings, researchers learned box embeddings by continuously optimizing the conditional probabilities of two entities forming a parent–child relationship, establishing box embeddings from a probabilistic perspective [17]. However, the existence of exact box-edge optimization conditional probabilities is difficult because gradient missing [31] makes disconnected box pairs challenging to optimize. In a recent study, BoxTaxo [20] proposed joint geometric and probabilistic views to learn box embeddings, which solves the gradient missing problem. Therefore the optimization of the box representation changes to continually optimizing the geometric and probabilistic views. Still, this continual minimization of the geometric and probabilistic losses results in the size of the box continuously decreasing until it is close to zero, but this produces meaningless boxes.

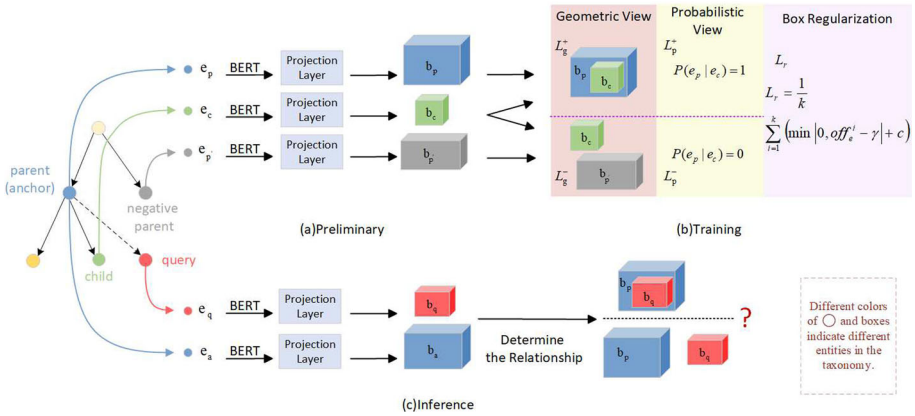


Fig. 3 The overview of PEB-TAXO

In our research, we introduced modified L1 regularization during the training process. This regularization penalizes the absolute values of the dimensions of the boxes, encouraging the boxes to move towards sparsity and have smaller values in each dimension while ensuring that they remain more significant than zero, thus preserving the existence of the boxes. As a result, the box embeddings possess a more stable and powerful representation ability. In this paper, we applied box embeddings to the taxonomy expansion task and achieved excellent results.

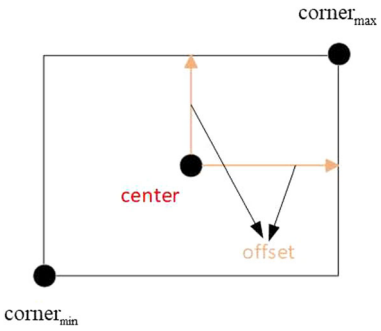
3 Methodology

This section will provide a detailed overview of the self-supervised model: PEB-TAXO, whose overall framework is illustrated in Fig. 3. The framework consists of three main components: (1) preliminary, where training samples are created from existing taxonomies, and entities from the samples are projected from natural language into the box embeddings through a two-stage projection process. (2) Training, where the model optimizes the boxes through joint views (geometric view, probabilistic view, and box regularisation) so that the boxes can represent the hierarchy more accurately. (3) Inference, PEB-TAXO projects the query encoding inside the box. Then, it finds suitable anchors for the queries by determining the inclusion relationship between the query box and candidate anchor boxes.

3.1 Preliminary

In the existing taxonomy, nodes represent concepts (entities). We treat each $< child, parent >$ pair as a positive sample and collect entities that are not ancestors of the child entity to form negative sample entity pairs, thus creating our training samples without annotated labels. Hence PEB-TAXO is a self-supervised model. In order to use the entities in the positive and negative samples to optimize the boxes in the training part, we need to first project these entities as boxes in the geometric space. For this purpose, PEB-TAXO uses a two-stage projection process. Specifically, entities are first encoded into numeric embeddings using an

Fig. 4 Box formation



entity encoder, and then the numeric embeddings are converted into boxes by a box projector. We describe this two-stage projection in detail below.

3.1.1 Entity Encoder

The pre-trained language models (PTM_s) have shown impressive performance across various natural language tasks [32]. Inspired by their success, we use PTM_s as the entity encoder to project entities into numerical embeddings. In this paper, we adopt BERT [33] as the entity encoder, and its representation is as follows:

$$n_i = Bert(e_i), n_i \in R^k \tag{1}$$

The meaning is: For the ith entity, it is transformed into an k-dimensional numerical embedding using BERT. Entities in taxonomy are typically curated by domain experts and have definition sentences, which can be converted into the input format of the BERT model: "[CLS] entity sentence [SEP]". We embed the "[CLS]" output in the last layer of the Bert model as the entity representation n_i. This representation encodes the contextual semantics of the entity.

3.1.2 Box Projector

After obtaining each entity's representation n_i, we project it into a box. A box embedding is a pair of vector embeddings that form a valid axis-aligned hyper-rectangle in k-dimensional space. A box can be defined by two points (vectors) [21], as shown in Fig. 4. Thus we use the center point $cen_i \in R^k$ as well as the offset $off_i \in R^k$ to determine the box $b_i \in R^k$, which is: $b_i \in (cen_i, off_i)$, where k is the dimension of the box embedding. It is worth noting that cen_i and off_i are vector embeddings obtained by projecting the entity representation n_i using two multi-layer perceptrons (MLP_s). The projections are of the form:

$$cen_i = MLP_{cen}(n_i), off_i = MLP_{off}(n_i) \tag{2}$$

Here, MLP_{cen} and MLP_{off} as the projectors for the center point cen_i and the offset off_i, respectively. After obtaining cen_i and off_i, we can derive the maximum corner point ($corner_{max}$) of the box: $l_i = cen_i + off_i, l_i \in R^k$ and the minimum corner point ($corner_{min}$): $s_i = cen_i - off_i, s_i \in R^k$. It is worth mentioning that $corner_{max}$ and $corner_{min}$ pairs can also define the box [34].

3.2 Training

After completing the preliminary step, we obtain the boxes for all entities in the training samples. However, these boxes are relatively basic and may not accurately capture the relationships between entities. Therefore, in this section, we optimize the box embeddings to represent the parent–child hierarchy accurately. In the following, we will show how to optimize the box embedding from three aspects: geometric view, probabilistic view, and box regularisation.

3.2.1 Geometric View

In this view, we first use boxes in geometric language to represent parent–child relationships. In geometric space, a k-dimensional box consisting of a center point and an offset is a k-dimensional hyper-rectangle. A $< child, parent >$ pair can be semantically interpreted as "the child is a part of the parent". Therefore, inside a geometric space, a child hyper-rectangle is fully contained by the parent hyper-rectangle and is part of the parent hyper-rectangle, i.e., a child entity is a kind of parent entity. Formally, the child box can be denoted as $b_c = (cen_c, off_c)$, one whose maximum and minimum corner points are: $l_c = cen_c + off_c, s_c = cen_c - off_c$. The parent box can be denoted as $b_p = (cen_p, off_p)$, whose maximum and minimum corner points are: $l_p = cen_p + off_p, s_p = cen_p - off_p$. Then, the inclusion relationship between the parent and child entities for $< e_c, e_p >$ can be expressed as:

$$l_c^i \leq l_p^i, s_c^i \geq s_p^i, \forall i \in \{1, 2, 3, \ldots, k\} \tag{3}$$

Here i is the ith dimension denoting the embedding. Based on this, we can derive a loss function L_g^+ to ensure that the boxes satisfy the geometric inclusion relationship between the parent–child pair:

$$L_g^+ = \frac{1}{k}\left[\sum_{i=1}^{k} max(0, l_c^i - l_p^i + \xi) + \sum_{i=1}^{k} max(0, s_p^i - s_c^i + \xi)\right] \tag{4}$$

where ξ is a hyperparameter that controls the geometric margin between the child and parent boxes and can span all k dimensions.

In contrast to the above, the $< child, negative\ parent >$ pairs in the negative samples, denoted as entity pair $< e_c, e_{p'} >$, are represented in the geometric space as child hyper-rectangles separated from their parent hyper-rectangles. To achieve this "disjoint" relationship, we force the intersection between the child box and the negative parent box to be empty. Specifically, for a box pair $< b_c, b_{p'} >$ to have an empty intersection $b_y = b_c \bigcap b_{p'}$, the maximum and minimum corners of b_y are represented as:

$$l_y = min(l_c, l_{p'}), s_y = max(s_c, s_{p'}) \tag{5}$$

An empty intersection means that the size of the intersection is less than or equal to zero in all k dimensions. Consequently, we formulate a loss function L_g^- that minimizes the offset off_y of the intersection set:

$$L_g^- = \frac{1}{k}\sum_{i=1}^{k} (off_y^i - \eta)^2 \tag{6}$$

Here η is the hyperparameter used to adjust the intersection margins. When we control this hyperparameter η to be less than zero, we force the separation of the child box from

the negative parent box. The intersection offset in the loss function can be found like this: $off_y = \frac{1}{2}(l_y - s_y)$.

3.2.2 Probabilistic View

Next, we will introduce how to represent parent–child relationships using box embeddings from a probabilistic perspective. We begin by defining a concept: Immediate family probability.

Definition 3.1 (*Immediate family probability*) The Immediate family probability $P(e_b|e_a)$ is the probability of the event: "For a given entity e_a, another entity e_b can be reached along an edge of a given length."

For a parent–child entity pair $< e_c, e_p >$ in the taxonomy, the Immediate family probability is $P(e_p|e_c) = 1$. This probability represents the likelihood that, given a child entity, we can find the corresponding parent entity by traversing along the edge connecting them. When a child entity has multiple parent entities, the Immediate family probability for all these parent entities is 1. On the other hand, for the entity pair $< e_c, e_{p'} >$ in the negative samples, the Immediate family probability is $P(e_{p'}|e_c) = 0$ because there is no direct connection from the given child node to the negative parent node. To accurately represent the parent–child hierarchy in the taxonomy using box embeddings, it is essential to satisfy these Immediate family probability conditions for both positive and negative entity pairs.

Box embeddings offer a natural way [35] to compute the Immediate family probability using geometric relations. We use the volume of the intersection of the parent and child boxes divided by the volume of the child box to represent the probability:

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

Here $Vol()$ denotes the box's volume, and the volume of a box is the product of the segments of this box in each dimension: $Vol(b) = \prod_{i=1}^{k}(l^i - s^i)$, where i is the dimension's index. Based on this, we can derive the probability loss function L_p^+ for each $< child, parent >$ pair:

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

Similarly, we can derive the probability loss function L_p^- for each $< child, negative\ parent >$ pair:

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

3.2.3 Box Regularization

Our negative loss functions are designed from both geometric and probabilistic perspectives, aiming to minimize the intersection between the child box and the negative parent box, i.e., negative geometric loss L_g^- and negative probabilistic loss L_p^-. However, if all the embedding dimensions of a box are close to zero, or if its volume approaches zero, minimizing these two losses may not be meaningful. In such cases, the learned box may not be well represented in the geometric space, leading to an ineffective representation of the parent–child hierarchy. To avoid this situation, we apply regularization to the boxes to restrict them from being too

small in all dimensions. Specifically, for boxes b_e that are projections of entities e, we limit the size of the box by using modified L1 regularisation for its offset off_e:

$$L_r = \frac{1}{k}\sum_{i=1}^{k}(min|0, off_e^i - \gamma| + c), \forall i \in \{1, 2, 3, \ldots, k\} \tag{10}$$

Here γ is the minimum length of the box that controls the box in each dimension, and c is a small positive constant that maintains the stability of the value. We penalize the absolute value of the box in all dimensions by this loss function, which drives the box towards sparsity, making the offset of the box even closer to the minimum length of the box. This restricts the box size from being too small in all dimensions while minimizing our negative loss function as much as possible.

3.2.4 Joint Loss

Finally, we combine a positive and negative geometric loss function, a positive and negative probabilistic loss function, and a modified L1 regularised loss function to train the boxes jointly. The final loss function is:

$$L = \omega(L_g^+ + L_g^-) + \rho(L_p^+ + L_p^-) + \kappa L_r \tag{11}$$

Here ω, ρ, and κ are hyperparameters controlling the contribution of the geometric loss function, the probabilistic loss function, and the regularised loss function.

3.3 Inference

During the inference phase, we aim to find suitable parent entities (anchors) from the existing taxonomy for a given query. Box embedding is more intuitive and natural in determining anchors than vector embedding, which uses the Euclidean distance measure when comparing entity relationships. We achieve this by examining the extent to which the query box is contained within the anchor boxes. We adopt the containment check method shown in Fig. 3c to implement the idea of finding anchors. Specifically, for a query e_q, we first project it into a box b_q, and then compare it with box b_a of all anchors e_a in turn. To determine whether the boxes b_q and b_a have a containment relationship, we define a containment index $Score$ based on the positions of their maximum and minimum corner points. If the following conditions are met: $l_q \leq l_a, s_q \geq s_a$, $Score = 1$, the anchor box contains the query box. Conversely, $Score = 0$, the anchor box is separated from the query box. The candidate anchor boxes that satisfy $Score = 1$ are the appropriate anchor boxes. We rank these candidate anchors based on the containment index, and the candidate anchors with containment indexes $Score = 1$ were all ranked higher than those with $Score = 0$. However, when the query box is contained by an anchor box that has a parent node and a grandfather node, it will also be contained by the boxes of all the ancestor nodes of this anchor. This means there will be more than one candidate anchor box with a containment index of 1. In this case, we select the lowest-level entity box among all the entity boxes containing the query box as the genuine anchor box. This box represents the finest granularity and provides a more precise description of the query box. Additionally, because all ancestor boxes contain this box, it has the smallest volume. Therefore, for candidate anchors with the same containment index, we perform a second ranking based on the volume of their boxes, to ensure that anchors with finer granularity are given higher priority.

Table 1 The statistics of the datasets for evaluation

Dataset	Terms	Edges	Layers
Environment	261	261	6
Science	429	452	8
Food	1486	1576	8

3.4 Logical Consistency Assurance

PBE-TAXO can effectively cope with logical inconsistencies. PBE-TAXO determines the parent–child relationship between entities by checking the containment relationship between boxes, i.e., A is a child of B when b_A is contained in b_B, in which case Vol(A)<Vol(B). When dealing with certain logical errors, such as A being predicted as a child of B while C is a parent of B, but A turns out to be a parent of C, Or when a child of an entity is predicted to be the parent of that entity, the model identifies these logical inconsistencies by comparing the relative positions and volume sizes of the boxes. It then corrects these logical inconsistencies, finds the true parent of each query, and ensures that the generated classification system remains logically correct.

4 Experiments

4.1 Datasets and Evaluation Metrics

In order to evaluate the effectiveness of PEB-TAXO in the taxonomy expansion task, we chose two commonly used English public datasets for our experiments, which are derived from the shared task of taxonomy construction in SemEval-2016 [36]. These two datasets correspond to the conceptual categorization of the domains Environment and Science, respectively, curated by domain experts, and their preset structure is a hierarchical structure about the parent–child relationships between terms (entities) of a given domain, where each entity is accurately categorized into the corresponding category, making these two datasets well suited for evaluating the performance of the taxonomy expansion model. Both datasets provide their definitions in addition to the taxonomic entities, the definitions are explanatory descriptions of these entities by domain experts, and we combine the entity names with their definitions as model inputs. In Table 1, we summarize their statistics. In addition, in both datasets, we randomly select the bottom 20% of the nodes in the taxonomy as a test set and keep the rest of such nodes in the training set. The model code can be found at: https://github.com/lizaozhou/PEB-TAXO.

It's worth noting that we opted not to utilize the Food dataset from SemEval-2016, and this decision is attributed to two key reasons: (1) PEB-TAXO is excessively complex, and due to limited experimental resources, the model can only be executed on the Environment and Science datasets. The extensive scale of the Food dataset, as detailed in Table 1, surpasses that of the Environment and Science datasets, rendering our experimental environment incapable of meeting the requirements for conducting experiments on the Food dataset. (2) The structural features of the Food dataset closely resemble those of the Environment and Science datasets, particularly the Science dataset, sharing the same number of layers and a similar ratio of nodes to edges. Consequently, we have chosen to exclusively conduct experiments on these two datasets to ensure reliable results within our constrained resources.

For all experiments, the following three metrics are used in this paper to evaluate our model and baseline: accuracy (ACC), mean reciprocal rank (MRR), and Wu & Palmer similarity (Wu&P).

- *Accuracy (ACC)* The precision of locating anchors for queries is defined as follows:

$$ACC = \frac{1}{k}\sum_{i=1}^{k}\mathbb{I}(z_i = \hat{z}_i) \tag{12}$$

- *MRR (Mean Reciprocal Rank)* The metric to measure the position of the actual anchors for queries in the ranked output is defined as follows:

$$MRR = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{rank(z_i)} \tag{13}$$

- *Wu & Palmer similarity (Wu&P)* The semantic similarity between predicted anchors and real anchors is calculated and defined as follows:

$$Wu\&P = \frac{1}{k}\sum_{i=1}^{k}\frac{2 \times depth(LCA(z_i, \hat{z}_i))}{depth(z_i) + depth(\hat{z}_i)} \tag{14}$$

Here "depth(.)" is the depth of the entity in the taxonomy, "LCA(.,.)" is the least common ancestor of the two inputs.

4.2 Baseline Algorithms and Parameter Settings

We use the above three evaluation metrics to measure the performance of PEB-TAXO and compare it to the following baseline:

- *TAXI* [32] This model is the best taxonomic induction method for the SemEval-16 task. It uses substring matching techniques to find significant strings in the text, then uses lexical patterns learned through extensive corpus data to establish hyponym–hypernym term pairs for the significant strings, and finally integrates these term pairs into a taxonomy.
- *HypeNet* [37] This method combines path analysis with distributed representation techniques, jointly modeling the distributional information and relation paths between term pairs using an LSTM-CNN model, thereby contributing to hypernym–hyponym relation detection.
- *Bert+MLP* [2] A pre-trained model-based hypernym detection method, which generates entity embeddings using the pre-trained language model Bert, and then feeds them into a multi-layer perceptron to predict the presence of hypernym relations between entities.
- *TaxoExpan* [1] TaxoExpan is an advanced self-supervised method that uses graph neural networks to encode the positional information in the taxonomy, and then identifies whether candidate terms are hypernyms of the query term through linear layers.
- *STEAM* [2] This model is a self-supervised method based on mini-paths, using mini-paths to model relationships between concepts in the taxonomy, and continuously optimizing the model through multi-view joint training to find suitable anchors for queries better.
- *BoxTaxo* [20] BoxTaxo uses box embeddings instead of traditional vector embeddings for entity representation, inferring relationships between entities by the degree of overlap between the intersection of boxes and the query box, and achieves excellent results in taxonomy expansion tasks.

We selected the same dataset for all baselines and the same evaluation metrics for model evaluation, and for all baseline models, we used the authors' published source code. For a fair comparison, we replaced the traditional word embedding approach in the TaxoExpan model with Bert embedding to perform better. For the other baseline models, we adjusted the parameters according to the original papers' specifications, aiming to achieve the best performance on the two datasets as reported in their respective papers, to participate in our comparison. Regarding the PEB-TAXO model, we set the learning rate to 0.00002, batch size to 100, dropout to 0.05, hidden layer size to 64, and MLP learning rate to 0.003, the small positive constant c that maintains the stability of the values to 0.000001, and the weighting of each individual loss to: $\omega = 1$, $\rho = 0.1$, $\kappa = 1$. The dimensionality of the box embeddings was adjusted within the range of 2 to 128, as the optimal dimensionality varied for different datasets. Additionally, for hyperparameters ξ, η, and γ, we tune them in a particular range, as described in Section 4.6. We typically set the training epoch to 100, and the model was implemented on PyTorch, utilizing an NVIDIA 3090Ti server.

4.3 Performance Comparison

Table 2 presents the comparison results of ACC, MRR, and Wu & Palmer for PEB-TAXO against all the baselines. We divided the baselines into three groups, and through observation and analysis, we drew the following conclusions:

(1) The first group includes TAXI, HypeNet, and Bert+MLP. Since the PEB-TAXO model is modeling simple hypernym–hyponym pairs as boxes, we compare with these baselines that use vector embeddings to represent pairs of hypernym–hyponym relationships. As shown in the table, PEB-TAXO significantly improves compared to all three methods, indicating that projecting entities as boxes performs well in taxonomy expansion tasks.
(2) The second part consists of TaxoExpan and STEAM. These two baselines use advanced structural summaries (local graph and mini-paths), obtaining better results than vector embeddings alone. Although PEB-TAXO does not model these complex structures, our results are still significantly more potent than these two baselines. However, PEB-TAXO no longer has a great advantage in MRR and Wu&P metrics compared to STEAM. This result suggests that advanced structures have a solid driving effect on taxonomy expansion, and thus using advanced structures to model boxes has excellent potential in taxonomy expansion tasks.
(3) The third part is only BoxTaxo. This baseline also uses box embeddings to model the hypernym–hyponym relations and performs better than most of the other baselines, so we compare PEB-TAXO with it separately. As shown in Table 2, our model demonstrates stronger capabilities. This result indicates that, using box embeddings with the incorporation of modified L1 regularization to constrain the size of boxes, and adopting the inference of inclusion relationship for reasoning about the hypernym–hyponym relations yields better performance in the taxonomy expansion task.
(4) Across all datasets, PEB-TAXO shows significant improvements. For instance, on the Environment dataset, PEB-TAXO outperforms the best baseline with increases of 10.0% in ACC, 5.8% in MRR, and 1.7% in Wu&P.

Table 2 Results on the three datasets

Dataset	Environment			Science		
Metric	ACC	MRR	Wu&P	ACC	MRR	Wu&P
TAXI	16.7	–	44.7	13.0	–	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
PEB-TAXO	48.1	52.9	77.1	42.4	50.4	73.1

Table 3 Ablation results

Datasets	Metrics	PT-R	PT-L	PT-I	PT-A	PEB-TAXO
Environment	ACC	23.5	42.3	44.2	38.5	48.1
	MRR	35.8	49.7	50.7	45.7	52.9
	Wu&P	63.2	73.8	76.2	73.3	77.1
Science	ACC	14.1	35.3	36.5	31.8	38.8
	MRR	25.8	46.0	48.9	43.7	49.2
	Wu&P	50.3	67.1	69.0	66.8	70.2

4.4 Ablation Tests

To validate the effectiveness of the components of the PEB-TAXO model, we conduct ablation tests. We investigated the impact of removing two components: the modified L1 regularisation that restricts the size of the boxes and the box inclusion inference method used for inferring anchors. Specifically, We designed four simplified models:

- *PT-R* Remove regularisation completely.
- *PT-L* Use the Mean Squared Error Loss Function (MSE) instead of the modified L1 regularisation.
- *PT-I* Use probabilistic perspective inference instead of box inclusion inference.
- *PT-A* Use MSE to limit box sizes and probabilistic perspective inference to find anchors for queries.

Table 3 gives the experimental results of PEB-TAXO on both datasets when the dimensions of the boxes are both 12.

By observing the experimental results of PT-R in Table 3, it is evident that completely removing the regularization leads to a significant decline in model performance. This discovery fully demonstrates the necessity of limiting the size of the box. However, the results of the PT-R model experiments are too poor to reflect the effect of modified L1 regularisation. To showcase this aspect, in other ablation experiments we regularized the box with MSE.

The experimental results of PT-L, PT-I, and PT-A models revealed the following: Firstly, the modified L1 regularization during the PEB-TAXO training phase and the box inclusion inference during the inference phase contribute differently to the model's performance. Although both components improve the performance of the model, it is clear that the modified L1 regularisation contributes more to the model. Secondly, we found that removing

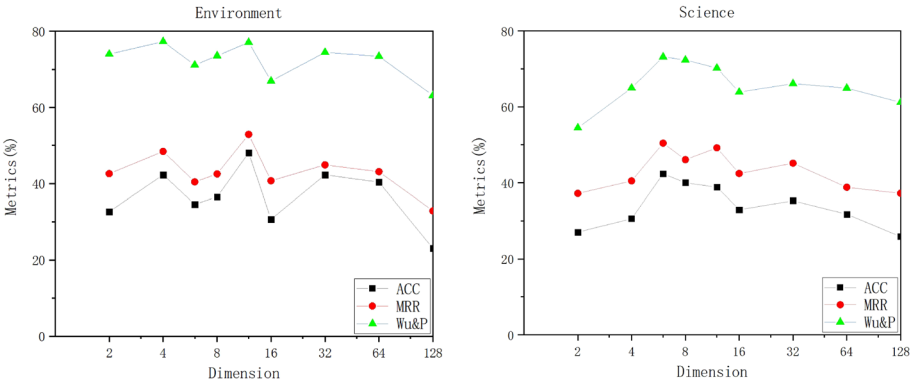


Fig. 5 Model performance metrics for different spatial dimensions

the box inclusion inference does not significantly affect PEB-TAXO. Whereas, after deleting the modified L1 regularisation, the performance of the model noticeably declines, and the three metrics show considerable fluctuations during the 100 epochs. This phenomenon indicates that using modified L1 regularization effectively constrains the box size within an ideal range, thus enhancing the model's performance. Finally, the performance of the model is significantly reduced when both box inclusion inference and modified L1 regularisation are removed. Therefore, our ablation experiments validated the effectiveness of these two components in PEB-TAXO.

4.5 Dimensional Experiment

In the high-dimensional geometric space, boxes are represented as hyper-rectangles, whose edge in each dimension is a line segment. In order to understand the effect of boxes trained from different dimensions on PEB-TAXO, we adjusted the number of box embedding dimensions between 2, 4, 6, 8, 12, 16, 32, 64, 128, and show the taxonomy expansion metrics for each dimension in Fig. 5. In both datasets, as the dimension increases, the metrics initially improve and gradually decline. We speculate that PEB-TAXO needs enough space to accommodate entities, so the dimensions cannot be too small. However, excessively large dimensions can lead to optimization difficulties, thereby increasing the challenge of taxonomy expansion. In our experiments, we observed that the model achieves its best performance for the environment dataset when the dimension is set to 12. On the other hand, for the science dataset, the optimal dimension is 6. These experimental results confirm our hypothesis.

4.6 Hyper-Parameter Studies

To investigate the impact of hyperparameters ξ, η, and γ on the performance of PEB-TAXO, we conducted experiments on both datasets with various values for these three hyperparameters. The results are shown in Fig. 6, and from observations and analysis, we draw the following conclusions:

(1) For parameter ξ, we experimented with nine values in the range of [0, 0.23]. As ξ increases, the metrics show an overall decreasing trend and a continuous downward pattern. In practice, ξ should not be too large, because training box embeddings involves

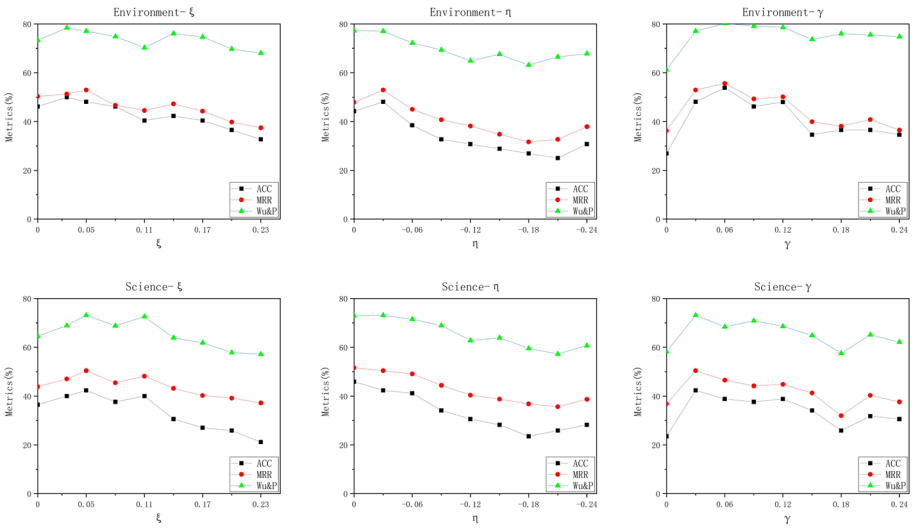


Fig. 6 Model performance metrics for different hyperparameters

continuously shrinking the intersection between parent and child boxes. When the geometric margin between the child and the parent boxes is too large, the process becomes more difficult, leading to insufficient model training and decreased performance.

(2) For hyperparameter η, we select nine values between [0,−0.24] for the experiment. From the observation in Fig. 6, it can be seen that η is sensitive to the edges, and the performance of the model declines as η decreases. η ensures the separation of the parent box from the child box in this value space. But when η is too small, the size of this nonexistent intersection becomes larger, which makes it difficult to minimize the negative geometric loss, and ultimately affects the model effect.

(3) Finally, for hyperparameter γ, nine values between [0,0.24] are taken for the experiment. In both datasets, as γ increases, the metrics generally increase and then decrease. Because the size of the box is limited as the minimum length of the box increases, thus avoiding many meaningless boxes, the performance of the model is naturally improved. This also proves the necessity of volume regularisation. However, when the length of the box is large, the difficulty of optimizing the box also increases, and the model effect is slightly reduced.

In conclusion, we selected hyperparameters $\xi = 0.05$, $\eta = -0.03$, and $\gamma = 0.03$. Under these settings, the model achieved excellent performance on both datasets.

5 Conclusion

This paper proposes a novel taxonomy expansion model PEB-TAXO that projects entities as boxes. Firstly, we use modified L1 regularisation to limit the size of boxes while optimizing the box embedding using geometric and probabilistic views, thus preventing the model from generating meaningless boxes, hence the robustness of the model is improved; Secondly, we introduce the box inclusion inference method, which uses the containment index to determine the existence of a containment relationship between two boxes, enabling us to

infer the anchor for a query in the taxonomy; Finally, we conduct extensive experiments on two public datasets, demonstrating the significant superiority of our model over all baseline models. Additionally, our ablation experiments confirm the effectiveness of each component. In the future, we plan to integrate mini-paths and local ego-graph graphs into the modeling of boxes, to further enhance the performance of extended models for taxonomy based on entities projecting boxes.

Acknowledgements This work was supported by the Science Fund for Outstanding Youth of Xinjiang Uygur Autonomous Region under Grant No. 2021D01E14.

References

1. Shen J, Shen Z, Xiong C, Wang C, Wang K, Han J (2020) TaxoExpan: self-supervised taxonomy expansion with position-enhanced graph neural network. In: Proceedings of the web conference 2020, pp 486–497
2. Yu Y, Li Y, Shen J, Feng H, Sun J, Zhang C (2020) Steam: self-supervised taxonomy expansion with mini-paths. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1026–1035
3. Huang J, Ren Z, Zhao WX, He G, Wen J-R, Dong D (2019) Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In: Proceedings of the twelfth ACM international conference on web search and data mining, pp 573–581
4. Zhang Y, Ahmed A, Josifovski V, Smola A (2014) Taxonomy discovery for personalized recommendation. In: Proceedings of the 7th ACM international conference on Web search and data mining, pp 243–252
5. Harabagiu SM, Maiorano SJ, Paşca MA (2003) Open-domain textual question answering techniques. Nat Lang Eng 9(3):231–267
6. Yin X, Shah S (2010) Building taxonomy of web search intents for name entity queries. In: Proceedings of the 19th international conference on World wide web, pp 1001–1010
7. Shen J, Wu Z, Lei D, Zhang C, Ren X, Vanni MT, Sadler BM, Han J (2018) Hiexpan: task-guided taxonomy construction by hierarchical tree expansion. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, pp 2180–2189
8. Snow R, Jurafsky D, Ng A (2004) Learning syntactic patterns for automatic hypernym discovery. Adv Neural Inf Process Syst 17
9. Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. In: COLING 1992 volume 2: the 14th international conference on computational linguistics
10. Chang H-S, Wang Z, Vilnis L, McCallum A (2017) Distributional inclusion vector embedding for unsupervised hypernymy detection. arXiv:1710.00880
11. Fu R, Guo J, Qin B, Che W, Wang H, Liu T (2014) Learning semantic hierarchies via word embeddings. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers), pp 1199–1209
12. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26
13. Wang S, Zhao R, Chen X, Zheng Y, Liu B (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, pp 3291–3304
14. Mao Y, Zhao T, Kan A, Zhang C, Dong XL, Faloutsos C, Han J (2020) Octet: online catalog taxonomy enrichment with self-supervision. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 2247–2257
15. Jiang M, Song X, Zhang J, Han J (2022) Taxoenrich: self-supervised taxonomy completion via structure-semantic representations. In: Proceedings of the ACM web conference 2022, pp 925–934

16. Liu Z, Xu H, Wen Y, Jiang N, Wu H, Yuan X (2021) TEMP: taxonomy expansion with dynamic margin loss through taxonomy-paths. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 3854–3863
17. Vilnis L, Li X, Murty S, McCallum A (2018) Probabilistic embedding of knowledge graphs with box lattice measures. arXiv:1805.06627
18. Dasgupta S, Boratko M, Zhang D, Vilnis L, Li X, McCallum A (2020) Improving local identifiability in probabilistic box embeddings. Adv Neural Inf Process Syst 33:182–192
19. Li X, Vilnis L, Zhang D, Boratko M, McCallum, A (2018) Smoothing the geometry of probabilistic box embeddings. In: International conference on learning representations
20. Jiang S, Yao Q, Wang Q, Sun Y (2023) A single vector is not enough: taxonomy expansion via box embeddings. In: Proceedings of the ACM web conference 2023, pp 2467–2476
21. Ren H, Hu W, Leskovec J (2020) Query2box: reasoning over knowledge graphs in vector space using box embeddings. arXiv:2002.05969
22. Onoe Y, Boratko M, McCallum A, Durrett G (2021) Modeling fine-grained entity types with box embeddings. arXiv:2101.00345
23. Shang J, Zhang X, Liu L, Li S, Han J (2020) Nettaxo: automated topic taxonomy construction from text-rich network. In: Proceedings of the web conference 2020, pp 1908–1919
24. Manzoor E, Li R, Shrouty D, Leskovec J (2020) Expanding taxonomies with implicit edge semantics. In: Proceedings of the web conference 2020, pp 2044–2054
25. Cocos A, Apidianaki M, Callison-Burch C (2018) Comparing constraints for taxonomic organization. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (Long Papers), pp 323–333
26. Agichtein E, Gravano L (2000) Snowball: extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries, pp 85–94
27. Jiang M, Shang J, Cassidy T, Ren X, Kaplan LM, Hanratty TP, Han J (2017) Metapad: meta pattern discovery from massive text corpora. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 877–886
28. Dash S, Chowdhury MFM, Gliozzo A, Mihindukulasooriya N, Fauceglia NR (2020) Hypernym detection using strict partial order networks. Proc AAAI Conf Artif Intell 34(05):7626–7633
29. Wang C, Fan Y, He X, Zhou A (2019) A family of fuzzy orthogonal projection models for monolingual and cross-lingual hypernymy prediction. In: The world wide web conference, pp 1965–1976
30. Vedula N, Nicholson PK, Ajwani D, Dutta S, Sala A, Parthasarathy S (2018) Enriching taxonomies with functional domain knowledge. In: The 41st international ACM SIGIR conference on research and development in information retrieval, pp 745–754
31. Li X, Vilnis L, Zhang D, Boratko M, McCallum A (2018) Smoothing the geometry of probabilistic box embeddings. In: International conference on learning representations
32. Qiu X, Sun T, Xu Y, Shao Y, Dai N, Huang X (2020) Pre-trained models for natural language processing: a survey. Sci China Technol Sci 63(10):1872–1897
33. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
34. Onoe Y, Boratko M, McCallum A, Durrett G (2021) Modeling fine-grained entity types with box embeddings. arXiv:2101.00345
35. Venn J (1880) I. On the diagrammatic and mechanical representation of propositions and reasonings. Lond Edinb Dublin Philos Mag J Sci 10(59):1–18
36. Bordea G, Lefever E, Buitelaar P (2016) Semeval-2016 task 13: taxonomy extraction evaluation (texeval-2). In: Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), pp 1081–1091
37. Shwartz V, Goldberg Y, Dagan I (2016) Improving hypernymy detection with an integrated path-based and distributional method. arXiv:1603.06076