Insert or Attach: Taxonomy Completion via Box Embedding

Anonymous ACL submission

Abstract

Taxonomy completion, enriching existing taxonomies by inserting new concepts as parents or attaching them as children, has gained significant interest. Previous approaches embed concepts as vectors in Euclidean space, which makes it difficult to model asymmetric relations in taxonomy. In addition, they introduce pseudo-leaves to convert attachment cases into insertion cases, leading to an incorrect bias in network learning dominated by numerous pseudo-leaves. To address these issues, we propose a novel taxonomy completion framework, called TAXBOX, which leverages the geometric properties of insertions and attachments in the box embedding space. By mapping concepts to box embeddings, TAXBOX can capture the complex relations between them, relying on the geometric connections between boxes. We also introduce a granular box constraint loss based on the hierarchy of the taxonomy, leading to more accurate concept mapping. Moreover, we design two geometric scorers, one for insertion and the other for attachment, which take into account the distinct behaviors of these two operations in the box embedding space. To balance the scores from the two scorers, we employ a dynamic ranking loss to adaptively adjust the magnitudes of the insertion score and the attachment score. Experiments on four real-world datasets show that TAXBOX significantly outperforms previous methods, yielding average performance improvements of 6.7%, 34.9%, and 51.4% in MRR, Hit@1, and Prec@1, respectively.

1 Introduction

Taxonomy, a critical knowledge graph with an "is-a" relationship, plays a vital role in information retrieval, recommendation systems, and question answering (Chatterjee and Das, 2022; Chuang and Chien, 2003; Kejriwal et al., 2022; Kerschberg et al., 2001; Suchanek et al., 2007; Huang et al., 2019; Yang et al., 2017; Yu et al., 2021). However, manual taxonomy enrichment is inefficient and costly due to the constant emergence of new concepts. To address the challenge of incorporating new concepts, taxonomy completion has been introduced, with new concepts either inserted as both parents and children or attached only as children (Jiang et al., 2022; Zhang et al., 2021; Wang et al., 2022; Zeng et al., 2021). This task goes beyond taxonomy expansion, which primarily treats new concepts as leaf nodes and tends to have limitations in downstream applications (Shen et al., 2020; Liu et al., 2021; Yu et al., 2020; Manzoor et al., 2020; Phukon et al., 2022; Jiang et al., 2023).

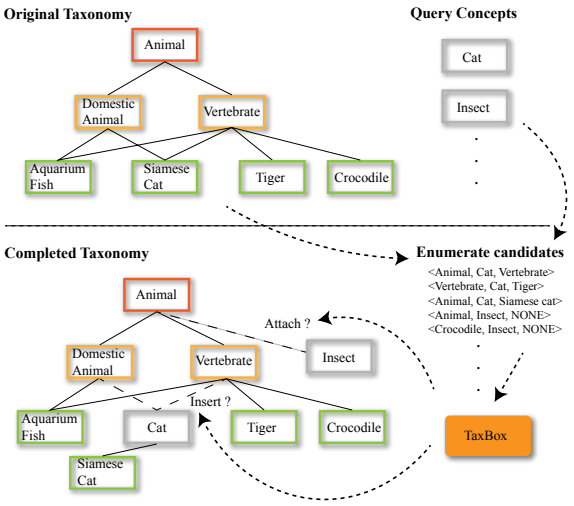

Figure 1: Example of taxonomy completion with our TAXBOX framework.

Taxonomy completion entails a more comprehensive incorporation of new concepts with two operations: insertion and attachment. For instance, in Figure 1, new query concepts such as *cat* and *insect* are added to the existing *animal* taxonomy. The process requires enumerating all possible candidate positions within the original taxonomy, including existing edges like <*Animal, Vertebrate*>

and implicit edges from each node to its descendants such as <*Animal, Tiger*>. Each candidate position is then paired with the query concept, and a confidence score is calculated. Finally, *insect* is attached as a child of *animal* and *cat* is inserted as a parent of *Siamese cat* and children of *Domestic Animal* and *Vertebrate* according to their confidences.

Recent research on taxonomy enrichment has examined various practical methods (Jiang et al., 2022; Zhang et al., 2021; Wang et al., 2022; Zeng et al., 2021). Nevertheless, all of these approaches embed concepts as vectors in Euclidean space, which makes them less capable of modeling the asymmetric relationship ("is-a") in taxonomy. BoxTAXO (Jiang et al., 2023) tried to employ box embedding, a representation method that can capture more prosperous and asymmetric relationships like inclusion, disjoint, and proximity among concepts through its geometric properties. However, this method is limited in real-world applications for its reliance only on the volume property, rendering it suitable only for the taxonomy expansion and even incapable of discerning optimal ancestor concepts and handling multiple parents during inference. Moreover, methods for taxonomy completion (Zhang et al., 2021; Wang et al., 2022) suffer from using a "pseudo-leaf" as a child node in attachment cases, leading to confusion in the matching. It is attributed that attachment cases often predominate due to leaf nodes' prevalence in real taxonomies. Therefore, learning too much about the pseudo-leaf in the attachment cases may reduce the network's perceptual ability for child nodes in the insertion cases.

To overcome these limitations, we present a novel framework for taxonomy completion called **TaxBox**, which is the first to apply box embedding to taxonomy completion. This approach adopts a structurally enhanced box decoder, representing concepts as box embeddings (Vilnis et al., 2018) encompassing the information of children, furnishing richer semantics. Most importantly, TaxBox combines two probabilistic scorers to unify the process of insertion and attachment in the box embedding space and incorporates both the volume and center closeness properties of box embedding. Such a design effectively exploits the fine-grained geometric attributes of box embeddings, circumventing the need for a pseudo-leaf and yielding optimal, feasible results during the ranking process. Additionally, we propose two novel training objectives, optimizing both box volume and position, and rectifying scorer numerical imbalances.

The specific contributions of this paper are outlined as follows:

- We introduce TaxBox, the first framework using box embedding for taxonomy completion with a structurally enhanced box decoder.
- We establish insertion and attachment scorers, obviating the need for pseudo-leaves and ensuring the determination of optimal results.
- We design box constraint loss, focusing on both volume and center closeness, and dynamic ranking loss, rectifying scorer numerical imbalance.
- Experimental outcomes from four datasets demonstrate our model's efficacy, achieving 6.7% MRR, 34.9% Hit@1, and 51.4% Prec@1 improvements over the previous methods.

2 Related work

Taxonomy Expansion and Completion. Taxonomy expansion, the process of attaching novel concepts into an existing taxonomy, has evolved over time with various approaches (Shen et al., 2018, 2020; Yu et al., 2020; Manzoor et al., 2020; Liu et al., 2021; Ma et al., 2021; Phukon et al., 2022; Jiang et al., 2023). Although effective, these methods have limitations in addressing real-world applications. Thus, Zhang et al. (2021) introduced taxonomy completion, a generalization that allows for the insertion of a concept as a parent to existing nodes, generating wider-reaching solutions. Subsequent research (Wang et al., 2022; Jiang et al., 2022; Zeng et al., 2021) sought to tackle this more challenging version of taxonomy expansion. Jiang et al. (2022) incorporated contextual embeddings into input embeddings, leveraging dual LSTMs to encode ancestor and descendant information (Staudemeyer and Morris, 2019). Meanwhile, Zeng et al. (2021) devised a generative strategy that concurrently generates concept names and classifies valid candidate positions. Wang et al. (2022) introduced the Quadruple Evaluation Network (QEN), which utilized pretrained language models (PLM) (Devlin et al., 2018; Sanh et al., 2019) to augment initial embeddings with semantically rich term representations. Arous et al. (2023) learns a position-

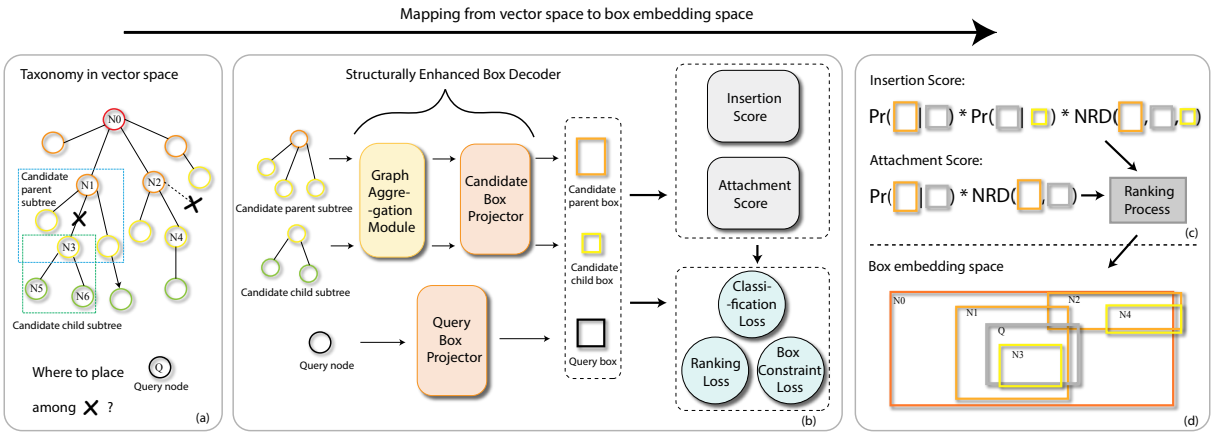

Figure 2: Overview of TAXBOX architecture. (a) The seed taxonomy tree with a query concept. (b) A structurally enhanced box decoder maps concepts among all the candidates and the query concept to the box embedding space. (c) Two probabilistic scorers calculate confidence of insertion or attachment for each candidate position. (d) Find the best position via ranking to complete the seed taxonomy with the novel concept in box embedding space.

enhanced node representation through anchor sets to better find the candidate.

Box Embedding. Box embedding represents a mapping technique that embeds concepts or objects within hyperplane boxes. Initially proposed by Vilnis et al. (2018), this approach employs probabilistic box lattices to encapsulate entities in knowledge graphs as *n*-dimensional rectangles. Subsequently, various studies have applied box embedding across diverse domains. For instance, Rau et al. (2020) predicted visual overlap in images, while Onoe et al. (2021) and Patel et al. (2021) focused on entity typing and multi-label classification, respectively. Moreover, Dasgupta et al. (2022) mapped words to capture set-theoretic semantics, and Hwang et al. (2022) and Messner et al. (2022) explored relation extraction and knowledge graph completion. These works highlight box embedding's suitability for nuanced semantic relationship modeling.

3 Preliminary

Box embedding (Vilnis et al., 2018; Chheda et al., 2021) refers to a mapping that represents a concept or object as a hyperplane box. A box $x = [x_m, x_M]$ is a hyperrectangle such that $x_m \in \mathbb{R}^d$ and $x_M \in \mathbb{R}^d$ where x_m and x_M represent the minimum and maximum endpoints of the box respectively along the d axis and $x_{m,i} \leq x_{M,i}$ holds for each axis $i \in \{1, 2, ..., d\}$. The center of box embedding is formulated as:

$$\text{Cen}(x) = \frac{x_M + x_m}{2} \quad (1)$$

There are two important operations: *Intersection* and *Volume* which are required for the calculation of the conditional probability of boxes' containment. Given two box embedding $x = [x_m, x_M]$, $y = [y_m, y_M]$, the *Intersection* of them is defined as follows:

$$\text{Inter}(x, y) = [\max(x_m, y_m), \min(x_M, y_M)] \quad (2)$$

where $min(\cdot,\cdot)$ and $max(\cdot,\cdot)$ in Equation 2 perform element-wise operations. Specifically, $min(a, b) = [min(a_1, b_1), ..., min(a_d, b_d)]$, and similarly for $max(\cdot,\cdot)$. The *Volume* is defined as:

$$\text{Vol}(x) = \prod_{i=1}^{d} \tau * \text{softplus}(\frac{x_{M_i} - x_{m_i}}{\tau}) \quad (3)$$
$$\text{softplus}(a) = \log(1 + \exp a)$$

where τ is a hyperparameter to adjust the smoothness. The probability of box x containing box y or the conditional probability of x given y is:

$$\Pr(x|y) = \frac{\text{Vol}(\text{Inter}(x, y))}{\text{Vol}(y)} \quad (4)$$

4 The TAXBOX Framework

In this section, we elaborate on the proposed TAXBOX framework, as shown in Figure 2. We begin by defining the problem in Section 4.1. Then, in Section 4.2, we introduce the structurally enhanced box decoder, which maps concepts into box embeddings with hierarchical information enhanced. Section 4.3 focuses on the discussion of two probabilistic scorers that evaluate the query and candidate boxes, providing attachment and insertion scores. Finally, in Section 4.4, we elucidate the learning objectives that contribute to improved optimization of box decoding and scorer balancing.

4.1 Problem Definition

A taxonomy is a directed acyclic graph and is defined as $\mathcal{T}^0 = (\mathcal{N}^0, \mathcal{E}^0)$ where each node $n \in \mathcal{N}^0$ represents a concept and each edge $\langle p, c\rangle \in \mathcal{E}^0$ represents the "is-a" relationship edge between concepts. Given a seed taxonomy $\mathcal{T}^0$ and a set of new concepts C, the definition of taxonomy completion is to construct a new taxonomy $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \mathcal{N}^0 \cup \mathcal{C}$ and $\mathcal{E}$ is updated by adding new edges among $\mathcal{C}$ and $\mathcal{N}^0$. To fulfill the task, all the candidate positions $\mathcal{P} = \{\langle p, c\rangle | \forall p \in \mathcal{N}^0, \forall c \in descendants(p)\}$ have to be evaluated given a novel concept $n \in \mathcal{C}$. The whole training paradigm follows self-supervised learning. For each node in the seed taxonomy, we pretend it to be a query and optimize it with a reconstructed taxonomy without the node.

4.2 Structurally Enhanced Box Decoder

The structurally enhanced box decoder includes a graph aggregation module to aggregate the hierarchical features from the ego subtree, as well as two box projectors map aggregated features and query embedding to box embedding space, respectively. An ego subtree of node n is defined as a tree only containing n and its one-hop children, denoted by $\mathbb{T}(n)$.

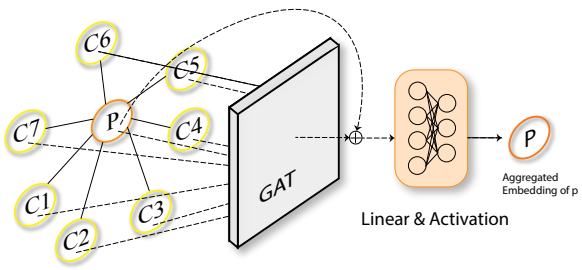

Figure 3: Details of Graph aggregation module.

For a query q and a possible candidate $\langle p,\ c\rangle \in \mathcal{P}$, we first obtain the embedding of each concept in the candidate along with their hierarchical information. As illustrated in Figure 3, we design a graph aggregation module to achieve this. The formulation is given by Equation 5:

$$F_k = \text{Lin}(\mathbb{R}(\text{GAT}(\mathbb{T}(k)) + \mathbb{T}(k))), k \in \{p, c\} \tag{5}$$

where F_k is the aggregated feature and $\mathbb{R}(\cdot)$ is a readout method, which implies that we only read out the root embedding of an ego subtree. *Lin* denotes a linear layer with activation. To effectively fuse more information from relevant child nodes, we opt for *GAT* (Graph Attention Network) (Veličković et al., 2018) to aggregate these trees in our implementation.

Next, two box projectors with identical Highway network(Srivastava et al., 2015) structure project aggregated features and query embedding to box embeddings, respectively, as formulated in Equation 6. To avoid potential conflicts arising from different latent spaces, we do not use a shared weight module for the aggregated parent/child features and query embedding.

$$\begin{aligned} B_q &= \text{QProjector}(F_q) \\ B_k &= \text{CProjector}(F_k),\ k \in \{p, c\} \end{aligned} \tag{6}$$

where F_q denotes query embedding and B_q, B_p, B_c represent the box embedding of query, candidate parent, and candidate child, respectively. QProjector is the query box projector, and CProjector is the candidate box projector.

4.3 Insertion and Attachment Scorer

To make the best use of the geometric properties of box embedding like volume and center closeness, we design insertion scorer and attachment scorer to separately give confidence corresponding to these two cases.

Insertion Scorer. Assumes that our model captures fine-grained semantic relationships between two concept boxes optimized by box constraint loss (Section 4.4). Given a query concept n, we first introduce its positive candidate set $C_{pos}(n) = \{\langle p,\ c\rangle | \forall p \in \mathbb{P}(n), \forall c \in \mathbb{C}(n)\}$ and negative candidate set $C_{neg}(n) = \{\langle p,\ c\rangle | \exists p \notin \mathbb{P}(n)\ \vee\ \exists c \notin \mathbb{C}(n)\}$ where $\mathbb{P}$ and $\mathbb{C}$ refers to the parents and children of a node. Note that $\mathbb{C}(n)$ can be an empty set. For a positive candidate pair, the parent box can reliably hold the child box, while two boxes within a negative pair are disjoint. The closer the pair is in position, the more overlapping their box embedding will be. Based on this, we propose an insertion scorer (S_I) that represents the likelihood of performing insertion into the candidate as follows:

$$\begin{aligned} S_I(B_q, B_p, B_c) = &\Pr(B_p|B_q) \cdot \Pr(B_q|B_c) \\ &\cdot \text{NRD}_t(B_q, B_p, B_c) \end{aligned} \tag{7}$$

where $\text{NRD}_t(\cdot, \cdot, \cdot)$ is the normalized reciprocal distance measuring the center closeness between the candidate parent and a query as well as that between the query and the candidate child. It is for-

mulated as:

$$\begin{aligned}\mathrm{RD}(B_q, B_{p_i}) &= \frac{1}{||\mathrm{Cen}(B_q) - \mathrm{Cen}(B_{p_i})||_2} \\ \mathrm{NRD}_p(B_q, B_p) &= \mathrm{softmax}_{i=1}^{n}(\mathrm{RD}(B_q, B_{p_i})) \\ \mathrm{NRD}_t(B_q, B_p, B_c) &= \mathrm{NRD}_p(B_q, B_p) \\ &\quad \cdot \mathrm{NRD}_p(B_q, B_c)\end{aligned} \tag{8}$$

where $\mathrm{softmax}_{i=1}^{n}$ represents applying $softmax$ along a mini-batch and B_{p_i} is a candidate in the mini-batch. $\mathrm{RD}(\cdot,\cdot)$ is the reciprocal distance, and $\mathrm{NRD}_p(\cdot,\cdot)$ only measures the closeness between the query and one side in the candidate.

Attachment Scorer. Similar to the insertion scorer, when faced with the scenario of a candidate pair with no child, an attachment scorer (S_A) is proposed. The attachment scorer is calculated as follows:

$$S_A(B_q, B_p) = \Pr(B_q|B_p) \cdot \mathrm{NRD}_p(B_q, B_p) \tag{9}$$

4.4 Multiple Learning Objectives

Classification Loss. The primary objective of our model is to determine the most suitable positions among all the candidate positions. We consider each candidate position as an independent category. Therefore, the problem can be reduced to a multi-label classification problem with a binary cross-entropy loss as:

$$\begin{aligned}\mathcal{L}_c = -\frac{1}{|\mathcal{B}|} &\sum_{(X_i, y_i)\in\mathcal{B}} y_i \log(S_k(X_i)) \\ &+ (1-y_i)\log(1 - S_k(X_i)), k \in \{I, A\}\end{aligned} \tag{10}$$

where $X_i = (B_{q_i}, B_{p_i}, B_{c_i})$, $\mathcal{B}$ refers to a mini-batch consisting of one positive sample and several negative samples, $y \in \{0, 1\}$ denotes whether the sample is positive or not. $S_k(k \in \{I, A\})$ means applying the insertion scorer if the candidate pair has both sides or the attachment scorer if it only has the parent side.

Box Constraint Loss. To better model the granularity of the "is-a" relationships amongst concepts using box embeddings, we focus on the geometric constraints originating from three properties of boxes: inclusion (l_{in}) and disjointness (l_{dis}) model the unidirectional relationships between two boxes, and centrality similarity (l_{cen}) facilitates scorers by obliging unrelated box pairs to assume orthogonal positions. Based on this, the loss functions for concept inclusion L_{in} and disjoint L_{dis} are as follows:

$$\begin{aligned}l_{in}(a,b) &= -\log \Pr(b|a) \\ l_{dis}(a,b) &= \max(0, \log(1-\gamma(a,b)) \\ &\quad - \log(1 - \Pr(a|b))) \\ l_{cen}(a,b) &= \max(0, \log(1-\gamma(a,b)) \\ &\quad - \log(1 - \mathrm{Cen}(a)\cdot\mathrm{Cen}(b))) \\ L_{in}(a,b) &= l_{in}(a,b) + l_{dis}(a,b) \\ L_{dis}(a,b) &= l_{dis}(a,b) + l_{dis}(b,a) + l_{cen}(a,b)\end{aligned} \tag{11}$$

The dynamic margin, $\gamma(a,b)$, between two concepts a and b, is adapted from the Wu&P similarity(Wu and Palmer, 1994) and modulates their semantic distance:

$$\gamma(a,b) = \alpha \cdot \frac{2 \times \mathrm{depth}(\mathrm{LCA}(a,b))}{\mathrm{depth}(a) + \mathrm{depth}(b)} \tag{12}$$

where $LCA(\cdot,\cdot)$ is the least common ancestor, $depth(\cdot)$ indicates the depth in the seed taxonomy, and α is a relaxation factor. By imposing constraints on volume(l_{in}, l_{dis}), position (l_{cen}), and distance ($S_{I/A}$), the optimization search space is effectively reduced.

Given a query box B_q, for a box $B_k(k \in \{p, c\})$ in a candidate, there are three possible scenarios: 1) B_q is contained within B_k. 2) B_k is contained within B_q. 3) both boxes are disjoint. When considering both sides of the candidate with a total of 6 possible cases, the box constraint loss is:

$$\begin{aligned}\mathcal{L}_b = &\frac{1}{|\mathcal{B}|}\sum_{(X_i, l_i)\in\mathcal{B}} l_{1i}\cdot L_{in}(B_{q_i}, B_{p_i}) \\ &+ l_{2i}\cdot L_{in}(B_{c_i}, B_{q_i}) \\ &+ l_{3i}\cdot L_{in}(B_{p_i}, B_{q_i}) \\ &+ l_{4i}\cdot L_{in}(B_{q_i}, B_{c_i}) \\ &+ (1-l_{1i})(1-l_{3i})\cdot L_{dis}(B_{q_i}, B_{p_i}) \\ &+ (1-l_{2i})(1-l_{4i})\cdot L_{dis}(B_{c_i}, B_{q_i})\end{aligned} \tag{13}$$

where $l_i = (l_{1_i}, l_{2_i}, l_{3_i}, l_{4_i})$ denotes whether the two sides of the candidate pair indeed contain the query concept or are contained by the query.

Ranking Loss. It's evident that the values of two scorers are numerically unbalanced, namely $S_I \leq S_A$ when considering the same candidate parent. In fact, there is no need for concern, as when a query is inserted into this candidate position, it is implicitly attached as a leaf. Our focus should be on guaranteeing $S_I(X_{pos}) \geq S_A(X_{neg})$ where

Method	MAG-CS							
	MR ↓	MRR	Hit@1	Hit@5	Hit@10	Prec@1	Prec@5	Prec@10
TaxoExpan	1523	0.099	0.004	0.027	0.049	0.017	0.023	0.021
ARBORIST	1142	0.133	0.008	0.044	0.075	0.037	0.038	0.033
TMN	639	0.204	0.036	0.099	0.139	0.156	0.086	0.060
QEN†	3960	0.147	0.017	0.062	0.097	0.076	0.054	0.042
TaxoEnrich*	5545	0.184	0.043	0.107	0.158	0.142	0.075	0.055
TAXBOX	**596**	**0.240**	**0.051**	**0.139**	**0.184**	**0.238**	**0.131**	**0.087**

Method	MAG-PSY							
	MR ↓	MRR	Hit@1	Hit@5	Hit@10	Prec@1	Prec@5	Prec@10
TaxoExpan	728	0.253	0.015	0.092	0.163	0.031	0.038	0.033
ARBORIST	547	0.344	0.062	0.185	0.256	0.126	0.076	0.052
TMN	212	0.471	0.141	0.305	0.377	0.287	0.124	0.077
QEN†	1778	0.293	0.103	0.150	0.206	0.103	0.059	0.042
TaxoEnrich*	2201	0.357	0.082	0.219	0.293	0.167	0.089	0.036
TAXBOX	**211**	**0.479**	**0.145**	**0.317**	**0.393**	**0.328**	**0.143**	**0.089**

Method	Wordnet-Verb							
	MR ↓	MRR	Hit@1	Hit@5	Hit@10	Prec@1	Prec@5	Prec@10
TaxoExpan	1799	0.227	0.024	0.095	0.140	0.036	0.029	0.021
ARBORIST	1637	0.206	0.016	0.073	0.116	0.024	0.022	0.018
TMN	1445	0.304	0.072	0.163	0.215	0.108	0.049	0.032
QEN*	2095	**0.331**	0.074	0.178	0.233	0.113	0.054	0.036
TaxoEnrich*	2873	0.320	0.069	0.168	0.229	0.106	0.052	0.035
TAXBOX	**1286**	0.330	**0.105**	**0.212**	**0.262**	**0.179**	**0.072**	**0.045**

Method	SemEval-Food							
	MR ↓	MRR	Hit@1	Hit@5	Hit@10	Prec@1	Prec@5	Prec@10
TaxoExpan	688	0.207	0.041	0.101	0.166	0.083	0.041	0.034
ARBORIST	700	0.129	0.013	0.053	0.088	0.027	0.022	0.018
TMN	559	0.211	0.037	0.113	0.160	0.074	0.046	0.032
QEN	353	0.313	0.070	0.176	0.234	0.146	0.074	0.049
TaxoEnrich†	305	0.348	0.113	0.247	0.290	0.230	0.100	0.063
TAXBOX	**281**	**0.359**	**0.132**	**0.264**	**0.295**	**0.318**	**0.127**	**0.071**

Table 1: Overall results on four taxonomy completion datasets. The ↓ denotes that the lower the metric is the higher performance the model has. Baselines are reported by Zhang et al. (2021) and Wang et al. (2022). * means our reproduction. † means our implementation on new datasets. We report the mean results of 5 runs.

the subscripts pos and neg indicate positive and negative samples, respectively. Consequently, for $k, k' \in \{I, A\}$, the ranking loss is strategically designed to circumvent this particular case.

$$\mathcal{L}_r = \frac{1}{|\mathcal{B}|} \sum_{X_i \in \mathcal{B}} \max(0, \gamma(X_{pos}, X_{neg}) + S_k(X_{neg}) - S_{k'}(X_{pos})) \tag{14}$$

Here, the dynamic margin compels $S_I(X_{pos})$ to be greater than $S_A(X_{neg})$ to a specific extent based on their structural similarity. The final loss combines all of the three losses mentioned above:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_b + \mathcal{L}_r \tag{15}$$

5 Experiments

5.1 Experiment Setup

Datasets. We assess TAXBOX's performance in taxonomy completion on four real-world datasets: two Microsoft Academic Graph subgraphs, *MAG-CS* and *MAG-PSY*, plus two WordNet subgraphs, *Wordnet-Verb* and *SemEval-Food*. Also, two public datasets from SemEval-16, *Science* and *Environment* are evaluated for taxonomy expansion. Further dataset details are available in Appendix A. Evaluation metrics consist of Mean Rank (MR), Mean Reciprocal Rank (MRR), Wu&P, Hit@k, and Prec@k, with elaboration in Appendix B.

Compared Methods. We select three recent SOTA taxonomy completion frameworks, Triplet Matching Network (TMN) (Zhang et al., 2021), QEN (Wang et al., 2022) and TaxoEnrich(Jiang

et al., 2022), and two taxonomy expansion frameworks, TaxoExpan (Shen et al., 2020) and ARBORIST (Manzoor et al., 2020), as baselines for the four completion datasets. Additionally, we compare BoxTAXO(Jiang et al., 2023) and TaxoExpan demonstrating TaxBox's superiority in taxonomy expansion. A further explanation is presented in Appendix C.

Implementation Details. The Adam optimizer was employed with a 0.001 learning rate and the ReduceLROnPlateau scheduler with a 10-epoch patience, training our model across all datasets for 100 epochs. Four attention heads were fixed with 0.1 dropout rate in GAT. The dynamic margin relaxation factor α was 0.5. The training and prediction smoothness factor τ were 10 and 20 respectively. Batch and negative sample size were set at 16 and 63, while box dimensions were set at 64 for *SemEval-Food*, 128 for *Wordnet-Verb* and *MAG-CS*, and 160 for *MAG-PSY*. Initial embeddings were the word2vec for the MAG datasets, fasttext for the Wordnet datasets, barring the PLM-based methods, and BERT embedding for two expansion datasets for fair comparison. All the experiments were conducted with one RTX3090.

5.2 Experimental Results

Table 1 demonstrates the superior performance of TaxBox in taxonomy completion datasets, reflecting average improvements of 6.7%, 34.9%, and 51.4% in MRR, Hit@1, and Prec@1. It outperforms prior SOTA models, such as QEN and TaxoEnrich, which utilize the pre-trained language models (PLM) to enhance the representation. It showcases TaxBox's performance when handling datasets with varied scales. TaxBox's efficacy originates from its box embedding's superior ability to capture asymmetric relationships among concepts and shows a significant improvement over conventional vector representations. PLM-based models like QEN, which lean on rich concepts' descriptions from various internet-based data sources, tend to induce noise, particularly when dealing with larger datasets with obscure, overlapping concepts. Similarly, TaxoEnrich's taxonomy-contextualized embeddings may reveal a variance in distribution between the training and testing phases, chiefly due to the test phase's exclusion of query-related information.

On *MAG-PSY* and *Wordnet-Verb* datasets, TaxBox outperforms in Hit@k and Prec@k metrics but has less exceptional MRR scores. A statistical analysis revealed that in *MAG-CS* and *SemEval-Food* datasets, the ratios of the maximum number of positive candidates in the training set to that in the test set are 2.5 and 1.5, respectively, whereas for *MAG-PSY* and *Wordnet-Verb*, the ratios are 14 and 11. It suggests the need for TaxBox to optimize for all the concept boxes under relatively relaxed conditions to accommodate numerous ground truth positions in the training set. This presents a challenge when identifying test queries with fewer ground truth positions, constricting MRR scores while showing significant improvements in other metrics.

5.3 Ablation Study

To assess the efficacy of our proposed learning objectives ($\mathcal{L}_r$, $\mathcal{L}_b$) and graph aggregation module, we performed ablation studies using *SemEval-Food* and *Wordnet-Verb* datasets (Table 2). The model's overall performance deteriorated when any component was removed, more noticeably so with $\mathcal{L}_b$. This is due to $\mathcal{L}_b$ explicitly constraining box location and volume, while $\mathcal{L}_r$ primarily balances the gap between scorers, which is implicitly addressed during the optimization process of $\mathcal{L}_c$. Despite that, $\mathcal{L}_r$ still yields a crucial 10% performance gain. The graph aggregation module demonstrated a significant improvement, underscoring its essential role in enhancing candidate feature enrichment.

Method	SemEval-Food		
	MRR	Hit@1	Prec@1
TaxBox w/o $\mathcal{L}_r$	0.346	0.104	0.250
TaxBox w/o $\mathcal{L}_b$	0.304	0.084	0.202
TaxBox w/o GAM	0.347	0.112	0.270
TaxBox w/o $\mathcal{L}_b$ &$\mathcal{L}_r$	0.285	0.079	0.189
TaxBox	0.359	0.132	0.318
Method	**Wordnet-Verb**		
	MRR	**Hit@1**	**Prec@1**
TaxBox w/o $\mathcal{L}_r$	0.316	0.097	0.165
TaxBox w/o $\mathcal{L}_b$	0.211	0.053	0.091
TaxBox w/o GAM	0.310	0.100	0.173
TaxBox w/o $\mathcal{L}_b$& $\mathcal{L}_r$	0.220	0.046	0.079
TaxBox	0.330	0.105	0.179

Table 2: Ablation study on SemEval-Food and Wordnet-Verb datasets. GAM means graph aggregation module.

5.4 How Two Scorers Work for Attachment and Insertion

Table 3 highlights the superior performance of TaxBox over *SemEval-Food* and *Wordnet-Verb* datasets in terms of attachment and insertion, compared to other methods. It excels in all attachment

metrics, emphasizing the aptitude of its scorer to utilize box embeddings' spatial aspects, while ignoring child boxes. For insertion, TAXBOX outperforms prevailing methods, indicating its scorer's accuracy in identifying optimal candidate positions considering overlap and center similarity. This confirms the effectiveness and necessity of our method, and the insufficiency of pseudo leaf introduction in prior methods.

Method	SemEval-Food			
	Attachment		Insertion	
	MRR	Hit@1	MRR	Hit@1
TMN	0.633	0.214	0.069	0.000
QEN	0.644	0.178	0.084	0.011
TAXBOX	**0.678**	**0.288**	**0.133**	**0.032**

Method	Wordnet-Verb			
	Attachment		Insertion	
	MRR	Hit@1	MRR	Hit@1
TMN	0.456	0.139	0.121	0.004
QEN	0.466	0.125	0.160	0.007
TAXBOX	**0.481**	**0.165**	**0.185**	**0.050**

Table 3: Performance in attachment and insertion cases.

5.5 How TAXBOX Solves the Limitation of BoxTAXO

Method	Environment		
	Prec@1	MRR	Wu&P
TaxoExpan	11.1	32.3	54.8
BoxTAXO	38.1	47.1	75.4
TAXBOX	**44.2**	**55.0**	**77.8**

Method	Science		
	Prec@1	MRR	Wu&P
TaxoExpan	27.8	44.8	57.6
BoxTAXO	31.8	45.3	64.7
TAXBOX	**44.7**	**54.3**	**81.3**

Table 4: The performance of TAXBOX on taxonomy expansion datasets. Baselines are reported by Jiang et al. (2023). *Please note that we have not scaled MRR by 10 and have applied a 100x scale to all results here.

Table 4 reveals that TAXBOX surpassed BoxTAXO in all metrics to show the TAXBOX's superiority over BoxTAXO. BoxTAXO's limitations largely stem from its simplification of taxonomies into sheer tree structures, resorting to containment or non-intersection. This approach engenders two primary concerns: 1) Hard boundaries inhibiting multiple parent nodes accommodation, and 2) unreliable inference criteria due to volume containment probability being the chief confidence score. Contrarily, TAXBOX mitigates these constraints with its soft margin-based constraints accommodating overlaps, and improves inference criteria with box center-position distance. Consequently, TAXBOX's predictions are more precise, and it capably processes nodes with multiple parents, outperforming BoxTAXO.

5.6 How Dynamic Margin Affects Box Constraint

Table 5 highlights the dynamic margin's efficiency in box constraint loss, in spite of comparable MRR results. Discrepancies in Hit@1 and Prec@1 across fixed margins accentuate the dynamic margin's superiority in accurately modeling inter-box relationships. While a 0.3 fixed margin in *SemEval-Food* might parallel its performance, determining the optimal margin remains challenging. Notably, the dynamic margin outperforms all fixed margins in *Wordnet-Verb*, further underscoring its adaptability.

Margin	SemEval-Food		
	MRR	Hit@1	Prec@1
0.1	0.357	0.107	0.256
0.3	0.355	0.121	0.291
0.5	0.352	0.104	0.250
dynamic	0.359	0.132	0.318

Margin	Wordnet-Verb		
	MRR	Hit@1	Prec@1
0.1	0.318	0.096	0.164
0.3	0.328	0.092	0.157
0.5	0.322	0.090	0.154
dynamic	0.330	0.105	0.179

Table 5: The results of different margins in the box constrain loss on two datasets.

5.7 How to set up TAXBOX

We discuss our choice for box dimensionality and the number of negative samples in Appendix D.

6 Conclusion

In this study, we present TAXBOX, a novel framework for taxonomy completion using box embeddings. Incorporating restricted box constraint loss, dynamic ranking loss, and two probabilistic scorers for attachment and insertion, TAXBOX employs a structurally enhanced box decoder, mitigating the need for pseudo leaves. Experiments on six real-world datasets demonstrate its effectiveness and performance. Future research could refine scorers without numerical imbalance and explore post-processing measures like reranking with LLM.

Limitations

The primary limitations of our proposed methods are as follows: (1) The numerical imbalance between the two scorers. Although we attempt to alleviate this issue by introducing a dynamic ranking loss, it remains an imperfect solution. Results shown in Table 3 indicate that tackling the insertion case in real-world practice is still challenging, despite TAXBOX achieving significant improvements compared to previous SOTA. A more practical scorer should be developed to address this. (2) In real-world applications, the quality of the initial embedding influences TAXBOX's performance to some extent. Even when we opt for a well-pretrained language model for encoding, the concept name and description have a considerable impact. Thus, a more adaptive training strategy is needed. For example, we could employ data augmentation techniques to generate multiple texts representing the same meaning and use a PLM to obtain an embedding set pointing to a specific concept. During training, we can then retrieve different embeddings to fit the network, consequently enhancing its generalization capabilities.

References

Ines Arous, Ljiljana Dolamic, and Philippe Cudré-Mauroux. 2023. Taxocomplete: Self-supervised taxonomy completion leveraging position-enhanced semantic matching. In *Proceedings of the ACM Web Conference 2023*, pages 2509–2518.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910.

Swarnali Chatterjee and Rajesh Das. 2022. Analysing and examining taxonomy and folksonomy terms in the hybrid subject device using machine learning techniques. *DESIDOC Journal of Library & Information Technology*, 42(3):154.

Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. 2021. Box embeddings: An open-source library for representation learning using geometric structures. *arXiv preprint arXiv:2109.04997*.

Shui-Lung Chuang and Lee-Feng Chien. 2003. Automatic query taxonomy generation for information retrieval applications. *Online Information Review*.

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)*, pages 2263–2276.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. New York, NY, USA. Association for Computing Machinery.

EunJeong Hwang, Jay-Yoon Lee, Tianyi Yang, Dhruvesh Patel, Dongxu Zhang, and Andrew McCallum. 2022. Event-event relation extraction using probabilistic box embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–244.

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*, pages 925–934.

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*, pages 2467–2476.

David Jurgens and Mohammad Taher Pilehvar. 2016. Semeval-2016 task 14: Semantic taxonomy enrichment. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1092–1102.

Mayank Kejriwal, Ravi Kiran Selvam, Chien-Chun Ni, and Nicolas Torzec. 2022. Local taxonomy construction: An information retrieval approach using representation learning. In *Social Media Analysis for Event Detection*, pages 133–161. Springer.

L. Kerschberg, Wooju Kim, and A. Scime. 2001. A semantic taxonomy-based personalizable meta-search agent. In *Proceedings of the Second International Conference on Web Information Systems Engineering*, volume 1, pages 41–50 vol.1.

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*, pages 3854–3863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*, pages 4182–4194.

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

Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7779–7787.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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)*, pages 2051–2064.

Dhruvesh Patel, Pavitra Dangati, Jay-Yoon Lee, Michael Boratko, and Andrew McCallum. 2021. Modeling label space interactions in multi-label classification using box embeddings. In *International Conference on Learning Representations*.

Bornali Phukon, Anasua Mitra, Ranbir Sanasam, and Priyankoo Sarmah. 2022. Team: A multitask learning based taxonomy expansion approach for attach and merge. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 366–378.

Anita Rau, Guillermo Garcia-Hernando, Danail Stoyanov, Gabriel J Brostow, and Daniyar Turmukhambetov. 2020. Predicting visual overlap of images through interpretable non-metric box embeddings. In *European Conference on Computer Vision*, pages 629–646. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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*, WWW '20, page 486–497, New York, NY, USA. Association for Computing Machinery.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, page 2180–2189, New York, NY, USA. Association for Computing Machinery.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. *arXiv preprint arXiv:1805.06627*.

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*, WWW '22, page 1008–1017, New York, NY, USA. Association for Computing Machinery.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138.

Shuo Yang, Lei Zou, Zhongyuan Wang, Jun Yan, and Ji-Rong Wen. 2017. Efficiently answering technical questions — a knowledge graph approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Wenhao Yu, Lingfei Wu, Yu Deng, Qingkai Zeng, Ruchi Mahindru, Sinem Guven, and Meng Jiang. 2021. Technical question answering across tasks and domains. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 178–186, Online. Association for Computational Linguistics.

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*, pages 1026–1035.

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, KDD '21, page 2104–2113, New York, NY, USA. Association for Computing Machinery.

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*, volume 35, pages 4662–4670.

A Dataset

We choose six real-world English datasets in different domains, four for taxonomy completion and two for taxonomy expansion. The statistical information about six datasets is shown in table 6.

- **Microsoft Academic Graph (MAG)** (Sinha et al., 2015) is a large, multi-disciplinary graph. The data in MAG includes information from a wide range of academic disciplines and includes more than 660 thousand scientific concepts and more than 700 thousand taxonomic relations. Following Zhang et al. (2021), we use subgraphs related to the computer science (**MAG-CS**) and psychology(**MAG-PSY**) domains. The initial embedding is a 250-dimension word2vec embedding trained by Zhang et al. (2021).

- **Wordnet** (Miller, 1995) is a large lexical database of English. Following (Wang et al., 2022) and (Zhang et al., 2021), we choose **Wordnet-Verb** (Jurgens and Pilehvar, 2016) and **SemEval-Food** (Bordea et al., 2015) which are extracted from wordnet. We employ 300-dimension fasttext embedding as our initial features following Zhang et al. (2021).

- **SemEval-16** we use two public datasets released from SemEval-16 task. Specifically, they are small-scaled taxonomy in the domains of **Environment** and general **Science**. And their initial embeddings are produced by a pre-trained bert(Devlin et al., 2018).

For *MAG-CS*, *MAG-PSY* and *Wordnet-Verb*, we randomly select 1,000 nodes for testing and 1,000 nodes for validation in each dataset, following the approach of Zhang et al. (2021). For *SemEval-Food*, we sample 10% of all the nodes for testing and another 10% for validation as done by Wang et al. (2022). For *Environment* and *Science*, we adopt the same protocol by Jiang et al. (2023). Subsequently, we reconstruct the seed taxonomy using the remaining nodes and add edges between the parent and child nodes of the test and validation sets to restore the fragmented taxonomy resulting from the dataset split.

Dataset	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert C\rvert$
MAG-CS	24,754	42,329	153,726
MAG-PSY	23,187	30,041	101,077
Wordnet-Verb	13,936	13,408	51,159
SemEval-Food	1,486	1,533	6,122
Science	344	354	344
Environment	209	209	209

Table 6: The statistics of six datasets. $\lvert\mathcal{N}\rvert$, $\lvert\mathcal{E}\rvert$, $\lvert C\rvert$ are the number of nodes, edges, and candidate positions, respectively.

B Evaluation Metric

All the methods as well as our model are ranking-based ones, so we use the ranking-based metric to evaluate performance. Supposing $rank(c_i)$ denotes the predicted rank of ground truth position given a query concept $c_i \in C$:

- **Mean Rank (MR)** mainly measures the average tail ranking level and we first calculate the average rank positions of each query and then average all the queries:

$$\text{MR} = \frac{1}{|C|}\sum_{i=1}^{|C|}\left(\frac{1}{M_i}\sum_{j=1}^{M_i}\text{rank}(c_i^j)\right) \quad (16)$$

  where M_i denotes the total number of ground truth positions of a query c_i and c_i^j denotes the jth prediction of c_i.

- **Mean Reciprocal Rank (MRR)** mainly measures the average head ranking level. Its form is similar to MR except that we get the reciprocal number of the ranks. Here we scale the reciprocal rank by 10 to amplify the difference.

$$\text{RR} = \frac{1}{M_i}\sum_{j=1}^{M_i}\frac{1}{\max(1, \text{rank}(c_i^j)/10)} \quad (17)$$

$$\text{MRR} = \frac{1}{|C|}\sum_{i=1}^{|C|}\text{RR} \quad (18)$$

- **Hit@k** measures the recall of a model which averages the true rank positions for all queries

in top k:

$$\text{Hit@k} = \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{M_i}\mathbb{1}(\text{rank}(c_i^j) \leq \text{k})}{\sum_{i=1}^{|C|} M_i} \tag{19}$$

- **Prec@k** measures the precision of the results and it sums the true rank positions of all queries in top k, divided by k times the total number of queries:

$$\text{Prec@k} = \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{M_i}\mathbb{1}(\text{rank}(c_i^j) \leq \text{k})}{\text{k} * |C|} \tag{20}$$

- **Wu&P**(Wu and Palmer, 1994) measures the structural similarity:

$$\text{Wu\&P} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{2 \times \text{depth}(\text{LCA}(a_i, b_i))}{\text{depth}(a_i) + \text{depth}(b_i)} \tag{21}$$

where a_i and b_i are the predicted top-1 result and the truth potision in taxonomy.

C Compared Methods

Here are the details of compared models:

- **TaxoExpan** (Shen et al., 2020): a state-of-the-art method in taxonomy expansion that utilizes a graph neural network to incorporate structural information.
- **ARBORIST** (Manzoor et al., 2020): a state-of-the-art framework for taxonomy expansion and it leverages heterogeneous edge semantics with a dynamic margin loss.
- **BoxTAXO** (Jiang et al., 2023): a state-of-the-art method using the property of conditional probability of box embedding for taxonomy expansion.
- **TMN** (Zhang et al., 2021): a state-of-the-art method for taxonomy completion that employs the channel-wise gate mechanism and auxiliary learning with multiple NTNs to evaluate partially positive candidate pairs beside positive pairs.
- **QEN** (Wang et al., 2022): a state-of-the-art model for taxonomy completion which utilizes a pre-trained language model to enhance the initial embedding with semantically rich term representation and enhance the performance with a sibling detector.
- **TaxoEnrich** (Jiang et al., 2022): a state-of-the-art model for taxonomy completion that leverages Taxonomy-Contextualized Embeddings and sibling matching modules.

D the Effect of Box Dimensionality and Negative Samples

We are also interested in how the box dimensionality and the number of negative samples affect the performance. Figure 4 shows the results of MRR, Hit@1 and Prec@1 when changing the box dimensionality from { 32, 64, 80, 128 } and the total number of samples from { 8, 16, 32, 64 }(where negative samples are { 7, 15, 31, 63 }) over two datasets.

Notably, it can be observed that for small datasets *SemEval-Food*, a dimension of 64 serves as a turning point. Dimensions below 64 exhibit a significant decline in overall performance. On the other hand, dimensions exceeding 64 reach a plateau, indicating that 64 is an appropriate dimension. Furthermore, increasing the dimension beyond 64 does not yield further performance improvements; instead, it leads to a decrease. This can be attributed to the fact that a dimension of 64 already satisfies the spatial constraints for all boxes in such a scale dataset. Larger dimensions introduce redundancy, thereby increasing the optimization difficulty. However, for *Wordnet-Verb*, it is worth noting that there is still some performance improvement observed after surpassing 64 dimensions. This discrepancy can be attributed to the larger dataset size and the initial quality of embeddings, which require more dimensions to effectively accommodate the information.

Regarding the setting of negative sample quantities, a general observation can be made that larger numbers of negative samples result in better performance on both datasets. However, it is crucial to acknowledge that an increased number of negative samples reduces the attention given to positive samples during the optimization process of the classification loss. Consequently, it becomes necessary to elevate the weight assigned to positive samples in calculations. Therefore, the steep decrease observed at the position of 16 is a consequence of equal weighting given to positive and negative

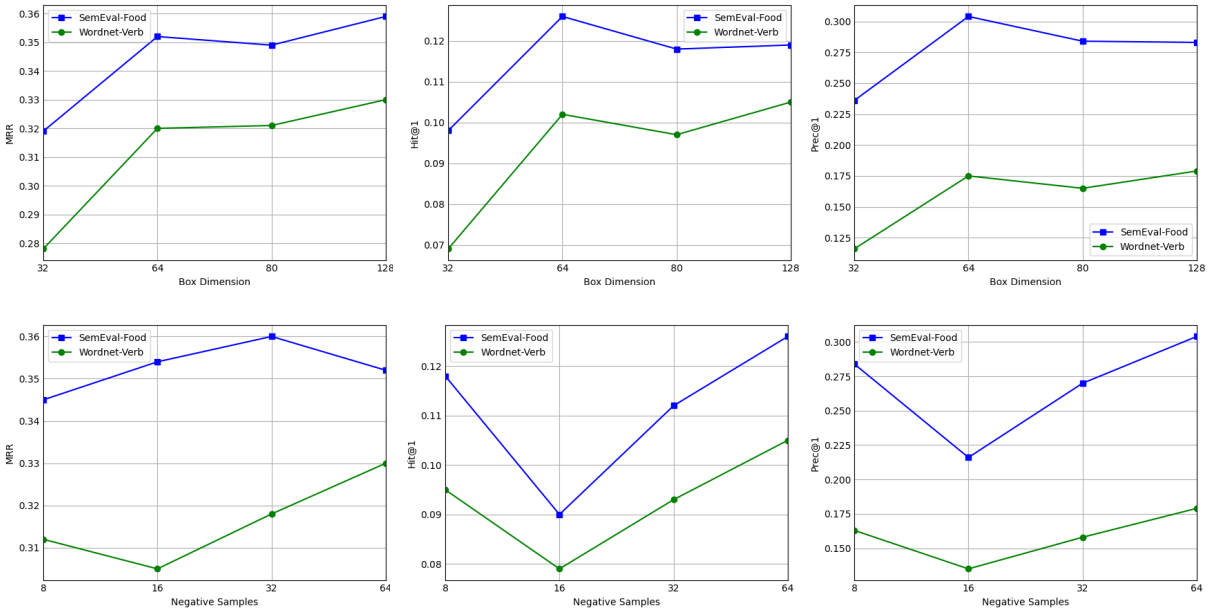

Figure 4: The effect of box dimensionality and the number of negative samples over three datasets.

samples in the experiment, while higher negative sample counts were assigned higher weights. This emphasizes the significance of appropriately adjusting the weight allocation to balance the impact of positive and negative samples during training.