HyperExpan: Taxonomy Expansion with Hyperbolic Representation Learning

Mingyu Derek Ma[1]   **Muhao Chen**[2*]   **Te-Lin Wu**[1*]   **Nanyun Peng**[1]

[1] Computer Science Department, University of California, Los Angeles
[2] Department of Computer Science and Information Sciences Institute,
University of Southern California

{ma,telinwu,violetpeng}@cs.ucla.edu   muhaoche@usc.edu

Abstract

Taxonomies are valuable resources for many applications, but the limited coverage due to the expensive manual curation process hinders their general applicability. Prior works attempt to automatically expand existing taxonomies to improve their coverage by learning concept embeddings in Euclidean space, while taxonomies, inherently hierarchical, more naturally align with the geometric properties of a hyperbolic space. In this paper, we present HyperExpan, a taxonomy expansion algorithm that seeks to preserve the structure of a taxonomy in a more expressive hyperbolic embedding space and learn to represent concepts and their relations with a Hyperbolic Graph Neural Network (HGNN). Specifically, HyperExpan leverages position embeddings to exploit the structure of the existing taxonomies, and characterizes the concept profile information to support the inference on unseen concepts during training. Experiments show that our proposed HyperExpan outperforms baseline models with representation learning in a Euclidean feature space and achieves state-of-the-art performance on the taxonomy expansion benchmarks.

1 Introduction

Taxonomy, a systematic categorization scheme, is an effective way to organize and classify knowledge (Härlin and Sundberg, 1998; Stewart, 2008). Taxonomies have been used to support many downstream applications such as content management in e-commerce (Wang et al., 2021b; Zhang et al., 2014), web search (Yin and Shah, 2010; Liu et al., 2019a), digital libraries (Yu et al., 2020), and NLP tasks (Yang et al., 2020; Hua et al., 2016; Yang et al., 2017). The curation of taxonomies mostly relies on human experts, which can be time-consuming and expensive, and hence suffer from limited coverage of the knowledge (Jurgens and
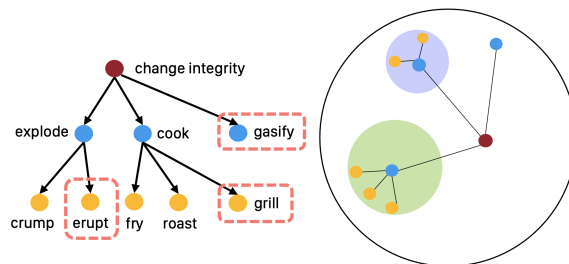

Figure 1: We show the taxonomy expansion task where red boxed concepts are newly attached concepts (left), and illustrate the representation of this taxonomy in a 2D Poincaré ball (right). Note that all the black edges have identical hyperbolic lengths.

Pilehvar, 2016). To alleviate this issue and handle constantly emerging new concepts, automating the taxonomy construction has attracted attentions from the research community (Wang et al., 2017). One type of such automated taxonomy curation is *taxonomy expansion*, which enriches an existing taxonomy to incorporate new and broader concepts. Specifically, the expansion of a taxonomy is performed as attaching new concept nodes to proper positions of a *seed* taxonomy graph, which is usually represented as a hierarchical tree (Vedula et al., 2018).

To systematically enrich a taxonomy graph, concept embeddings are firstly learned by structurally characterizing the concepts in the existing taxonomies, which are then used to match the embeddings of query concepts for the expansion. Prior works learn the concept embeddings with local structural features, such as edge semantic representation (Manzoor et al., 2020) and graph neural networks (GNN) (Shen et al., 2020). However, as a concept can lead to multiple subconcepts, the sizes of taxonomies expand exponentially with respect to their levels. The Euclidean embedding space, where existing works commonly build upon, fails to account for this property. In contrast, a *hyperbolic space* (Nickel and Kiela, 2017; Sarkar, 2011),

---

*Equal contributions.

4182

where the circumference of a negative-curved space grows exponentially with regard to the radius as illustrated in Figure 1, can better capture such special characteristics of taxonomies.

In this paper, we present HYPEREXPAN, a taxonomy expansion framework based on hyperbolic representation learning, that: (1) better preserves the taxonomical structure in a more expressive hyperbolic space, (2) effectively characterizes concepts by exploiting sparse neighborhood information beyond standard parent-child relations (Aly et al., 2019; Le et al., 2019), and (3) improves inference precision and generalizability by leveraging pretrained distributional features. [1]

Specifically, HYPEREXPAN incorporates two types of features to exploit the structural presentation of a taxonomy: a relative positional embedding of a node depending on its relation to the anchor node, and an absolute positional embedding defined by its depth within a taxonomy. HYPEREXPAN first constructs an *ego subgraph* around the potential attaching candidate concepts, *i.e.* the anchor concepts, and then leverages a hyperbolic graph neural network (HGNN) to obtain the *anchor concept embeddings*. A parent-child matching score for the attachment is subsequently produced by comparing both the anchor and query concept embeddings in the same hyperbolic space.

We evaluate HYPEREXPAN on WordNet and Microsoft Academic Graph datasets. Experiments show that the learned hyperbolic concept embeddings achieve better expansion performance than the Euclidean counterpart, outperforming the state-of-the-art models. We also perform ablation studies to demonstrate the effectiveness of each component and the design choice of HYPEREXPAN. Our contributions are summarized as follows: (1) We present an effective and generalizable taxonomy expansion framework via hyperbolic representation learning. (2) We introduce methods to incorporate pretrained distributional features and taxonomy-specific information in the hyperbolic GNN design. (3) We show that our framework achieves state-of-the-art performance on expanding four large real-world taxonomies.

2 Preliminaries

We introduce preliminaries about hyperbolic geometry and then define the task.

2.1 Hyperbolic Geometry

Hyperbolic space is a non-linear space with constant negative curvature as opposed to Euclidean space which has zero curvature. The curvature of a space measures how a geometric object deviates from a flat plane.[2] Specifically in this work, we mainly employ the following two models of hyperbolic geometry (Beltrami, 1868; Cannon et al., 1997): the Poincaré ball model and the Lorentz model, with some intermediate projective operations defined by the Klein model (see § 3.1).

There are several essential vector operations required for learning embeddings in a hyperbolic space, including: (1) computing the distance between two points, (2) projecting from a hyperbolic space to a Euclidean space, and vice versa, (3) adding and multiplying matrices, (4) concatenating two vectors, and (5) transformation among hyperbolic models. These necessary algebraic operations are summarized in Table 1.

For each point $x \in \mathcal{H}^n$ in the hyperbolic space, we denote the associated tangent space centered around x as $T_x \mathcal{H}^n$, which is always a subset of the Euclidean space. We make use of the *exponential map* $\exp_x : T_x \mathcal{H}^n \to \mathcal{H}^n$ and *logarithmic map* $\log_x : \mathcal{H}^n \to T_x \mathcal{H}^n$ to project points in the hyperbolic space to the local tangent space for precise approximation, and vice-versa. Setting the origin (north pole) of the hyperbolic space as the center, we can obtain a common tangent space across different manifolds as long as they are of the same dimension and modeled by the same hyperbolic model using $\log_0$ and $\exp_0$ projection. And hence, we can use $\log$ and $\exp$ to perform the projection within a neural network that has a mixture of hyperbolic and Euclidean layers.

The addition and matrix multiplication operations in Poincaré model are based on Möbius transformation (Ungar, 2001; Ganea et al., 2018; Gülçehre et al., 2019), which are defined in Table 1. In the Lorentz model, we utilize the tangent space to perform matrix multiplication and parallel transport to perform the addition (Chami et al., 2019).

For concatenating two hyperbolic vectors, we perform a generalized version of the concatenation operation (Ganea et al., 2018; López and Strube, 2020) to prevent the resulting vector from being

---

[2]Here we assume a unit hyperbolic space (curvature $= -1$) in this section.

4183

	Poincaré Ball	Lorentz Model
Distance	$d(x,y) = \cosh^{-1}\left(1 + 2\frac{\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right)$	$d(x,y) = \text{arcosh}\left(- <x,y>_{\text{L}}\right)$
Exponential Map	$\exp_x(v) = x \oplus \left(\tanh\left(\frac{\lambda_x \|v\|}{2}\right)\frac{v}{\|v\|}\right)$	$\exp_x^K(v) = \cosh\left(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}}\right)x + \sqrt{K}\sinh\left(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}}\right)\frac{v}{\|v\|_{\mathcal{L}}}$
Logarithmic Map	$\log_x(y) = \frac{2}{\lambda_x}\text{artanh}(\| -x \oplus y\|)\frac{-x \oplus y}{\|-x \oplus y\|}$	$\log_x^K(y) = d_{\mathcal{L}}^K(x,y)\frac{y+\frac{1}{K}\langle x,y\rangle_{\mathcal{L}} x}{\|y+\frac{1}{K}\langle x,y\rangle_{\mathcal{L}} x\|_{\mathcal{L}}}$
Addition	$x \oplus y = \frac{\left(1+2\langle x,y\rangle+\|y\|^2\right)x+\left(1-\|x\|^2\right)y}{1+2\langle x,y\rangle+\|x\|^2\|y\|^2}$	$x^H \oplus^K y := \exp_{x^H}^K\left(P_{\mathbf{o}\to x^H}^K(y)\right)$
Matrix Multiplication	$M \otimes x = \tanh\left(\frac{\|Mx\|}{\|x\|}\tanh^{-1}(\|x\|)\right)\frac{Mx}{\|Mx\|}$	$M \otimes^K x^H := \exp_{\mathbf{o}}^K\left(M\log_{\mathbf{o}}^K\left(x^H\right)\right)$

Table 1: Distance metrics and arithmetic operations in Poincaré and Lorentz models.

out of the manifold, as shown below:

$$\text{concat}(\mathbf{x}_1, \mathbf{x}_2) = M_1 \otimes \mathbf{x}_1 \oplus M_2 \otimes \mathbf{x}_2 \oplus b$$

where M_1, M_2 and b are parameters.

The Poincaré ball model $\mathcal{B}$, the Klein model $\mathcal{K}$ and the hyperboloid/Lorentz model $\mathcal{L}$ are used in our work, and we perform different computation on different models. These models are isometric isomorphic. Given a node $\boldsymbol{x} = [x_0, x_1, \cdots, x_n] \in \mathcal{L}$, the bijections between node on Lorentz model and its corresponding mapped node on Poincaré ball $\boldsymbol{b} = [b_0, b_1, \cdots, b_{n-1}] \in \mathcal{B}$ are (Cannon et al., 1997; Iversen and Birger, 1992):

$$p_{\mathcal{L}\to\mathcal{B}}(\boldsymbol{x}) = \frac{[x_1, \cdots, x_n]}{x_0 + 1}$$
$$p_{\mathcal{B}\to\mathcal{L}}(\boldsymbol{b}) = \frac{\left[1 + \|\boldsymbol{b}\|^2, 2\boldsymbol{b}\right]}{1 - \|\boldsymbol{b}\|^2}$$

The bijections between $\boldsymbol{x}$ and its mapped node on the Klein model $\boldsymbol{k} = [k_0, k_1, \cdots, k_{n-1}] \in \mathcal{K}$ are:

$$p_{\mathcal{L}\to\mathcal{K}}(\boldsymbol{x}) = \frac{[x_1, \cdots, x_n]}{x_0}$$
$$p_{\mathcal{K}\to\mathcal{L}}(\boldsymbol{k}) = \frac{1}{\sqrt{1 - \|\boldsymbol{k}\|^2}}[1, \boldsymbol{k}]$$

2.2 Taxonomy Expansion

In this work, a taxonomy is mathematically defined as a directed acyclic concept graph $\mathcal{T} = (\mathcal{N}, \mathcal{E})$, where each node $n \in \mathcal{N}$ represents a concept, and each directed edge $n_p \to n_c \in \mathcal{E}$ denotes a parent-child relation in which n_p and n_c is a pair of hierarchically related concepts (*e.g.* `change integrity` → `explode`). Given an existing taxonomy $\mathcal{T}^0 = (\mathcal{N}^0, \mathcal{E}^0)$, the goal of the taxonomy expansion is to attach a set of new concepts $\mathcal{C}$ to $\mathcal{T}^0$, *expanding* it to $(\mathcal{N}^0 \cup \mathcal{C}, \mathcal{E}^0 \cup \mathcal{R})$ where $\mathcal{R}$ are new edges whose children must be $c \in \mathcal{C}$.

An illustration of the taxonomy expansion is as shown in Figure 1, where the query nodes (new concepts) are *attached* to the proper positions depending on the surrounding *anchor* nodes (existing concepts). Following the settings of prior works (Shen et al., 2020; Zhang et al., 2021), we consider attaching different query concepts independently from each other to simplify the problem. Each concept in $\mathcal{N}^0 \cup \mathcal{C}$ has its profile information, *i.e.* concept definitions, concept names, and related articles etc. (See § 4.1 for more details.)

3 HYPEREXPAN

We propose HYPEREXPAN, a taxonomy expansion framework based on hyperbolic geometry and GNNs. As shown in Figure 2, HYPEREXPAN consists of the following main steps: 1) initial concept feature generations utilizing the profile information (§ 3.1). 2) encoding query and anchor concept features with hyperbolic (graph) neural networks (§ 3.2). 3) computing the query-anchor embedding matching scores for attaching query concepts to proper anchor positions (§ 3.3). We will describe each step in details and how to train the matching model (§ 3.4) in the following sections.

3.1 Initial Concept Features

We mainly leverage two types of profile information to obtain the initial concept (either in query or existing taxonomy) features: the name and the definition sentences of a concept. We firstly embed the two profile information by applying an average pooling over the word embeddings of each profile word, and then take the mean of the two embedded profile information to produce the fixed-dimension initial concept embedding. Our framework does not require the initial word embeddings to be defined in a specific geometry, and thus it can be either Euclidean, such as fast-
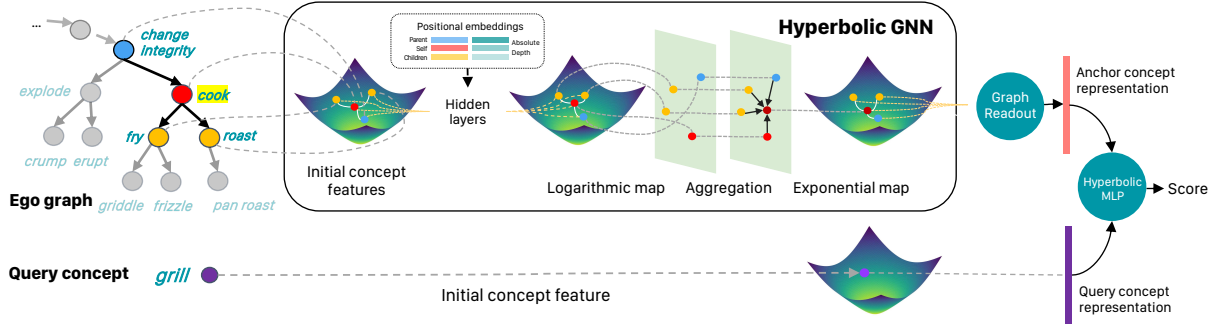
4184

Figure 2: HYPEREXPAN's model design. Red node is the anchor concept and the highlighted sub-tree is the ego graph of the anchor node. The intermediate flat surface is the tangent space based on the anchor node.

Text (Bojanowski et al., 2017), or hyperbolic, such as Poincaré GloVe (Tifrea et al., 2019), which embeds words in a Cartesian product of hyperbolic spaces. Note that since Poincaré GloVe is defined in hyperbolic space, the aforementioned mean operation can no longer be the usual Euclidean average since it may produce results that are out of the manifold. Instead, we use Einstein midpoint method (Gülçehre et al., 2019) to perform the average pooling. Denote the token embeddings as e_i and N as number of tokens in a sentence, the midpoint can be computed as:

$$\mu = \frac{\sum_{i=1}^{N} \gamma_i e_i}{\sum_{i=1}^{N} \gamma_i}$$

where $\gamma_i = \frac{1}{\|x_i\|^2}$ denotes the Lorentz factors. Einstein midpoint has the most concise form with the Klein coordinates (Gülçehre et al., 2019), therefore we project Poincaré embeddings to the Klein model $\mathcal{K}$ to calculate the midpoint, and then project the results back to the Poincaré model. We project the initial concept embeddings to the hyperbolic space $\mathcal{H}$ initialized by the following network design and used as the network input.

3.2 Anchor Concept Representation

We learn a parameterized model to encode anchor nodes a_i, taking the initial concept features x_{a_i} as inputs, and output the hyperbolic embedding vectors $\mathbf{o}_{a_i}$. We use HGNN to model the concepts in a hyperbolic space and exploit the structured representation of a taxonomy. We leave the basics of Euclidean Graph Convolutional Networks in Appendix A.

HGNN performs the neighbor aggregation operation in a hyperbolic space $\mathcal{H}$, which can be a Lorentz model $\mathcal{L}$ or a Poincaré model $\mathcal{B}$, following

corresponding numerical operations defined in § 2.1. Note that the standard neighbor aggregation operation in (Euclidean) GNN may lead to manifold distortion when embedding graphs with scale-free or hierarchical structure (Deza and Laurent, 2009; Bachmann et al., 2020).

The first layer of an HGNN maps initial node features (can be on a Euclidean or any hyperbolic spaces) to $\mathcal{H}$, followed by a series of cascaded HGNN layers. At each layer, the HGNN performs four operations in the following order: 1) transforming node features to messages in a predefined hyperbolic space, 2) transforming messages to the tangent space for each node, 3) performing neighborhood aggregation on the tangent space, and 4) projecting updated tangential node embeddings to hyperbolic space $\mathcal{H}$. In this work, our HGNN design is based on the hyperbolic graph convolutional network (Chami et al., 2019).

Ego Graph. To encode anchor concepts with an HGNN, an *ego graph* centered around anchor concept a_i is firstly constructed, where all nodes on such a graph is bounded by a certain edge distance.

Positional Features. To further exploit the structural presentation of a taxonomy, we incorporate the relative and absolute positional embeddings as inputs to an HGNN layer. With respect to a given center node, the neighbors of such node can be of one of the following three relative positions: parent, child, and self. Denote $pr_c(i)$ as the relative position of node i if the center node is c, we have the relative positional embedding as: $\mathbf{x}_{pr_c(i)}$.

Motivated by You et al. (2019); Wang et al. (2019), we equip the HGNN model with the position-awareness by incorporating an absolute position embedding. We define an *absolute position*, $pa(i)$, of a node i as its *depth* (*i.e.* level w.r.t

the root) within the entire taxonomy. Since the expansion task does not break the structure of the existing taxonomy, such position encoding is fixed for a given node. The depth-dependent position embedding is defined as $\mathbf{x}_{pa(i)}$. And hence, the overall inputs to each HGNN layer is a concatenation of the original node embeddings and the two taxonomy-specific features [3]:

$$\mathbf{x}_i^{\ell,\mathcal{H}} \leftarrow \mathbf{x}_i^{\ell,\mathcal{H}} \,\|\, \mathbf{x}_{pr_c(i)}^{\mathcal{H}} \,\|\, \mathbf{x}_{pa(i)}^{\mathcal{H}}$$

Note that the positional embeddings are initialized and then projected to hyperbolic space following Table 1, while the concatenation is as described in § 2.1. $\mathbf{x}_i^{0,\mathcal{H}}$ is the initial concept feature obtained following § 3.1.

Feature Transformation. At layer l, we transform the embedding vectors produced by the previous layer $\mathbf{x}_i^{\ell-1,H}$ to message $\mathbf{h}_i^{\ell,H}$ by applying a hyperbolic linear transformation:

$$\mathbf{h}_i^{\ell,\mathcal{H}} = \left(W^\ell \otimes^K \mathbf{x}_i^{\ell-1,\mathcal{H}}\right) \oplus^K \mathbf{b}^\ell$$

where $\otimes^K$ and $\oplus^K$ denotes multiplication and addition in hyperbolic space $\mathcal{H}$ with curvature K.

Neighborhood Aggregation. The neighborhood aggregation encapsulates neighboring features to update the center node. To enable an importance-weighted aggregation and for the simplicity to reuse Euclidean operations to derive the attention scores, we firstly apply a logarithmic mapping to project the messages to a tangent space. Let i be the center node and j be one of its neighbor nodes, we compute an attention weight w_{ij} by applying an Euclidean MLP to the concatenated tangential representations of the two messages following:

$$w_{ij} = \sigma_{j \in \mathcal{N}(i)} \left(\text{MLP}\left(\log_{\mathbf{o}}^K\left(\mathbf{h}_i^{\mathcal{H}}\right) \,\|\, \log_{\mathbf{o}}^K\left(\mathbf{h}_j^{\mathcal{H}}\right)\right)\right).$$

where σ is a softmax function over all neighbors $\mathcal{N}(i)$. The center node embedding is thus obtained by a weighted sum of the neighboring tangential embeddings. Finally, we apply an exponential mapping to project the aggregated tangential center node embedding to the hyperbolic model $\mathcal{H}$ as:

$$\text{AGG}^K\left(\mathbf{h}_i^{\mathcal{H}}\right) = \exp_{\mathbf{h}_i^{\mathcal{H}}}^K \left(\sum_{j \in \mathcal{N}(i)} w_{ij} \log_{\mathbf{h}_i^{\mathcal{H}}}^K\left(\mathbf{h}_j^{\mathcal{H}}\right)\right).$$

Note that for a better local hierarchical approximation, an *independent* local tangent space is created

[3]Superscript $^{\mathcal{E}}$ and $^{\mathcal{H}}$ indicate the node feature is in Euclidean space and hyperbolic space respectively.

for each center node i during the neighborhood aggregation, instead of using the tangent space of the hyperbolic origin (*i.e.* using $\exp_{\mathbf{h}_i^{\mathcal{H}}}^K$ and $\log_{\mathbf{h}_i^{\mathcal{H}}}^K$ instead of $\exp_{\mathbf{o}}^K$ and $\log_{\mathbf{o}}^K$). The curvature K of a hyperbolic model can either be a fixed number or a learnable parameter, where our experiments show that learned K tends to yield better performance. The update rule of the embedding of node i can thus be defined as:

$$\mathbf{x}_i^{\ell,\mathcal{H}} = \sigma(\text{AGG}^{K_{\ell-1}}\left(\mathbf{h}_i^{\ell-1,\mathcal{H}}\right)),$$

and we concatenate the updated node embedding with taxonomy-specific features and use as input for next layer. Finally we obtain the ego graph representation using the finalized node embeddings via a weighted readout function for the 1-hop neighbor nodes. Given G as 1-hop ego graph, $pr_{a_i}(j)$ as node j's relative positions (parent, child or self) related to center node a_i, $\alpha_{pr_{a_i}(j)}$ as the weight for node-type, then the concept representation for anchor node a_i is:

$$\mathbf{o}_{a_i} = \sum_{j \in G} \frac{\log\left(1 + \exp\left(\alpha_{pr_{a_i}(j)}\right)\right)}{\sum_{j' \in G} \log\left(1 + \exp\left(\alpha_{pr_{a_i}(j')}\right)\right)} \mathbf{x}_i^{\ell,\mathcal{H}}.$$

3.3 Matching Module

Given the initial concept features x_{q_i} of a query concept q_i, we obtain the query concept representation $\mathbf{o}_{q_i}$ by projecting x_{q_i} to the hyperbolic space $\mathcal{H}$ using the exponential mapping function (if $x_{q_i} \in \mathcal{E}$) or hyperbolic model transformation (if x_{q_i} is in other hyperbolic models other than $\mathcal{H}$) defined in § 2.1. Note that the hyperbolic spaces used to obtain the anchor and query concept representations need to be consistent.

After obtaining the hyperbolic embedding representation for each query concept $\mathbf{o}_{q_i} \in \mathcal{H}$ and each anchor concept $\mathbf{o}_{a_i} \in \mathcal{H}$, $\mathbf{o}_{q_i}$ and $\mathbf{o}_{a_i}$ are concatenated with hyperbolic operations, and then we feed the concatenated vector to an HNN. We construct hyperbolic multi-layer perceptron (MLP) based on the operations defined in (Ganea et al., 2018), and a one-layer HNN is defined as:

$$f^{\text{HNN}}(x) = \varphi^{\otimes^K}(M \otimes^K x \oplus^K b)$$

where $M \in \mathcal{R}^{m \times n}$ and $b \in \mathcal{H}^m$ are learnable parameters. Since b lies in a hyperbolic space, its update during training needs to be calibrated to remain in the proper manifold. $\varphi^{\otimes}$ is the element-wise non-linearity, where $\otimes^K$ and $\oplus^K$ denotes multiplication

and addition in hyperbolic space, respectively, under the curvature K. Note that HNN is equivalent to a Euclidean MLP if K is set to 0, *i.e.* the embedding space is not *curved*.

3.4 Learning and Inference

We train the HYPEREXPAN framework with a metric learning paradigm by utilizing the existing taxonomies as the training resources.

Training Data Construction. The data pairs that are used to train the matching module is generated in a self-supervised manner following Shen et al. (2020). We only consider *exact* parent-child node pairs on the seed taxonomy $\mathcal{T}^0$ as the positive samples, *i.e.* there exists a *direct* edge $\langle a_i, q_i \rangle$. For each query node q_i, we randomly sample N other nodes (without its immediate children) on the seed taxonomy to form negative training instances $\langle n_i^1, q_i \rangle, \langle n_i^2, q_i \rangle, ..., \langle n_i^N, q_i \rangle$. Anchoring at node q_i, the positive and negative samples form a single group of training instances $\mathbf{X}_i = \{\langle a_i, q_i \rangle, \langle n_i^1, q_i \rangle, ..., \langle n_i^N, q_i \rangle\}$. We repeatedly apply this operation on each edge of the seed taxonomy to construct our training data $\mathbb{X} = \{\mathbf{X}_1, ..., \mathbf{X}_{|\mathcal{E}^0|}\}$.

Learning Objective. We adopt InfoNCE loss (Oord et al., 2018) as the main training objective:

$$\mathcal{L}(\Theta) = -\frac{1}{|\mathbb{X}|} \sum_{\mathbf{X}_i \in \mathbb{X}} \left[\log \frac{f(a_i, q_i)}{\sum_{\langle n_i^j, q_i \rangle \in \mathbf{X}_i} f(n_i^j, q_i)} \right]$$

where $j \in [0, 1, 2, ..., N]$ and n_i^0 is the positive sample a_i. The InfoNCE loss is essentially a cross entropy loss which identifies the positive pairs (items in the numerator) among all the possible candidates (items in the denominator).

Inference. During the inference time, for each new query concept (unseen from the seed taxonomy) q_i, we compute the matching scores between the query concept q_i and every candidate anchor nodes $a_{candidate} \in \mathcal{N}^0$ in the existing taxonomy $\mathcal{T}^0$. We then rank these anchor nodes by the matching score to create a ranked list of length $|\mathcal{N}^0|$ for deciding where to attach such new concept.

4 Experiments

We evaluate the HYPEREXPAN and its variants on four large-scale real-world taxonomies utilized by Shen et al. (2020) and Zhang et al. (2021).

4.1 Experimental Setup

Datasets. Following Shen et al. (2020); Zhang et al. (2021), we take WordNet 3.0 and 1000 domain-specific concepts defined in SemEval-2016 Task 14 Benchmark dataset (Jurgens and Pilehvar, 2016), where only hypernym-hyponym relations are considered. WordNet thereof is separated into the verb version *WordNet-Verb* and the noun version *WordNet-Noun*. We also use subgraphs of the Field-of-Study Taxonomy in Microsoft Academic Graph (Sinha et al., 2015) containing descendants of "psychology" and "computer science" node and refer as *MAG-PSY* and *MAG-CS*.

Dataset	# Nodes	# Edges	Depth
WordNet-Verb	13,936	13,408	13
WordNet-Noun	83,073	76,812	20
MAG-PSY	23,187	30,041	6
MAG-CS	24,754	42,329	6

Table 2: Dataset statistics.

More detailed statistics of each dataset are in Table 2. For each dataset, 1000 leaf nodes are randomly sampled as query nodes as the validation set, and another 1000 leaf nodes form the test set. Since these validation and testing nodes are all leaf nodes, only minimum changes are required to make the remaining taxonomy still a valid one without introducing non-existed edges. For WordNet-Verb and WordNet-Noun, we generate the initial concept features following § 3.1. We assume each concept has only one name and we induce the concept name from the WordNet synset name. For MAG-PSY/CS, we use 250-d in-domain concept name word embeddings provided by Shen et al. (2020) trained using skipgram model on paper abstract corpus. Since we do not have access to the source profile information, we cannot obtain initial concept features as designed in § 3.1. As a result, we cannot run two RoBERTa-base baseline models introduced in the following section on the MAG-PSY/CS dataset.

Evaluation Metrics. We follow prior studies (Zhang et al., 2021; Shen et al., 2020; Manzoor et al., 2020) to report several widely-used ranking metrics: *MeanRank (MR), Mean Reciprocal Rank (MRR)*,[4] *Recall @ K* and *Precision @ K*.

---

[4] We report the MRR numbers scaled by 10 following previous works to amplify the performance difference.

Baseline Models. We compare HYPEREXPAN with the following strong baseline models:

- **RoBERTa-base Zero-shot**: we use RoBERTa-base as feature extractor to obtain initial embeddings as described in § 3.1 without fine-tuning

- **RoBERTa-base FT**: the above design but update the LM's parameters

- **Hyperbolic MLP**: we concatenate initial features of query and anchor concepts and feed into a two-layer hyperbolic MLP

- **GCN** (Kipf and Welling, 2017): HYPEREXPAN's design but use Euclidean GCN to update node embeddings in ego graph of the anchor concept, use fastText to obtain initial concept features, and use Euclidean MLP as the matching module

- **GAT** (Velickovic et al., 2018): the above method but use GAT to update node embeddings.

We compare HYPEREXPAN with the following state-of-the-art taxonomy expansion frameworks:

- **TaxoExpan** (Shen et al., 2020) uses GCN and GAT to get node embeddings of ego networks of anchor nodes and average all node embeddings to get anchor concept representation. But the ego network only includes direct children and parent of the anchor concept. They also inject relative positional embeddings to GNN.

- **ARBORIST** (Manzoor et al., 2020) combines global and local taxonomic information to explicitly model heterogeneous and unobserved edge semantics.

- **TMN** (Zhang et al., 2021) uses auxiliary scorers to capture various fine-grained signals including query to hypernym and query to hyponym semantics and introduces a channel-wise gating mechanism to retain task-specific information.

4.2 Experimental Results

The overall experimental results are shown in Table 3. We introduce our implementation details and hyperparameter settings in Appendix B.

Among ARBORIST, TaxoExpan and TMN, TMN achieves the strongest result consistently. Note that TMN is trained on taxonomy completion task and only perform inference on taxonomy expansion task. Anchor node representations are learned coupled with different potential children of the query concept which provides fine-grained signals. TaxoExpan performs better than ARBORIST

showing the importance of neighborhood information. Experiments using RoBERTa-base on two WordNet datasets indicate that RoBERTa language model falls drastically behind in this context-independent task. Since the profile sentences are very short and the task is more rely on common-sense rather than context understanding, language models cannot benefit from contextualized representation, which consolidates the observation by Liu et al. (2020). We can observe Hyperbolic MLP is worse than GNN models since it does not use neighborhood profile information. Hyperbolic MLP outperforms ARBORIST with a large margin on all datasets. The comparison between GCN and GAT indicates that attentive aggregation is more helpful with the sparse neighborhood representation. If we compare HYPEREXPAN with GCN and GAT, we can observe that the expressiveness of the hyperbolic space leads to a large performance increase (9.5% and 6.9% recall@10 increase on MAG-PSY and WordNet-Verb and MRR scaled by 10 increase ranging from 0.013 to 0.076). Overall, HYPEREXPAN consistently outperforms all models across four datasets except MR for WordNet-Noun.

To further help understand the contribution of different incorporated techniques, we present a series of ablation study results in Table 4. Specifically, we have the following observations:

According to lines 1-3, the trainable curvature learns fine-grained suitable manifold setting and lead to almost 2% recall@10 improvement (lines 1-3). Replacing the default Lorentz model with Poincaré model notably hinders the performance which can be explained by Lorentz model's numerical stability since the distance function of the Poincaré model contains fraction (Chami et al., 2019; Peng et al., 2021). We replace Poincaré GloVe initial word embedding with fastText in line 3 and the result shows that Poincaré GloVe contains better feature for our task.

We explore different choices of neighborhood aggregation in lines 4-7. We observe that 2-hop neighborhood aggregation leads to improvement over 1-hop in terms of recall@10 and precision@1 (line 5). Adding descendant of the anchor node supports with better characterization of nodes (line 6). However, we observe a noticeable drop when we further add sibling nodes into the aggregation neighborhood (line 7). The potential reason is that the sibling nodes are very diverse, and thus are not closely related to the anchor node.

Model	MR ↓	MRR ↑	Recall % ↑ @1	@5	@10	Precision % ↑ @1	@5	@10	MR ↓	MRR ↑	Recall % ↑ @1	@5	@10	Precision % ↑ @1	@5	@10
			WordNet-Verb (Candidates #: 11,936)								WordNet-Noun (Candidates #: 81,073)					
ARBORIST	608.7	0.280	10.8	24.0	27.7	6.7	4.8	3.2	1095.1	0.435	16.5	28.4	34.1	16.8	5.8	3.5
TaxoExpan	502.8	0.439	12.4	28.2	35.2	12.4	5.6	3.5	649.6	0.562	19.7	38.2	47.4	20.1	7.8	4.8
TMN	465.0	0.479	14.9	31.6	37.9	13.2	6.4	4.0	**501.0**	0.595	20.7	40.5	50.1	21.1	8.3	5.1
RoBERTa-base 0-shot	2132.8	0.172	4.3	10.1	12.6	4.3	2.0	1.3	25235.4	0.158	13.7	15.7	15.7	14.0	3.2	1.6
RoBERTa-base FT	1535.7	0.155	2.4	6.4	9.9	2.4	1.3	1.0	27748.2	0.148	5.9	13.7	13.7	6.0	2.8	1.4
Hyperbolic MLP	617.4	0.419	10.5	25.6	33.7	10.5	5.1	3.4	869.6	0.502	18.1	33.6	41.7	18.5	6.9	4.3
GCN	456.9	0.445	10.9	27.2	34.5	10.9	5.4	3.5	684.1	0.563	20.9	39.8	47.3	21.3	8.1	4.8
GAT	471.7	0.449	11.6	28.7	35.6	11.6	5.7	3.6	640.7	0.585	22.3	40.9	49.7	22.7	8.3	5.1
HYPEREXPAN	**400.8**	**0.517**	**15.0**	**32.8**	**42.7**	**15.0**	**6.6**	**4.3**	573.6	**0.607**	**23.9**	**42.1**	**52.5**	**24.4**	**8.6**	**5.4**
			MAG-PSY (Candidates #: 21,187)								MAG-CS (Candidates #: 22,754)					
ARBORIST	119.9	0.722	21.0	48.4	62.9	25.8	12.5	7.7	284.7	0.602	15.1	38.9	49.4	24.6	12.6	8.0
TaxoExpan	68.5	0.775	26.1	56.9	69.5	33.8	14.7	9.0	189.8	0.661	15.9	42.9	55.4	25.8	13.9	9.0
TMN	73.0	0.781	25.8	58.7	70.5	33.4	15.2	9.1	160.5	0.667	16.0	43.1	56.3	26.0	14.0	9.1
Hyperbolic MLP	74.1	0.739	21.8	51.4	64.9	28.2	13.3	8.4	101.4	0.650	13.7	38.0	53.4	22.3	12.4	8.7
GCN	51.4	0.742	23.8	52.5	64.3	30.8	13.6	7.4	90.3	0.653	14.5	39.6	53.3	23.6	12.9	8.7
GAT	48.6	0.751	23.6	52.4	65.8	30.5	13.5	8.5	92.2	0.676	15.9	41.9	56.0	25.9	13.6	9.1
HYPEREXPAN	**38.4**	**0.827**	**28.8**	**63.0**	**75.3**	**37.2**	**16.3**	**9.7**	**74.4**	**0.689**	**16.1**	**44.6**	**58.0**	**26.1**	**14.5**	**9.4**

Table 3: Overall experimental results. Directions (pointing up or down) of arrows indicate better performance of the metrics. MRR metrics are scaled by 10 to amplify the performance difference.

#	Model	MRR ↑	Rec ↑ @10	Prec ↑ @1
1	w/o trainable curvature	0.490	40.8	14.4
2	Poincaré i/o Lorentz model	0.494	39.8	13.0
3	fastText i/o Poincaré GloVe	0.494	41.0	15.2
4	anchor + parent + children	0.506	42.2	15.0
5	#4 + anchor's ancestors	0.505	42.5	15.5
6	#5 + anchor's descendants	0.517	42.7	15.0
7	#6 + anchor's siblings	0.502	41.7	14.5
8	w/o Relative Pos Emb	0.497	40.8	13.0
9	w/o Absolute Pos Emb	0.503	41.2	14.3
10	w/o both Positional Emb	0.482	38.8	12.5
	HYPEREXPAN	0.517	42.7	15.0

Table 4: Experimental results for ablation studies on WordNet-Verb. By default, we use trainable curvature, Lorentz hyperbolic model, Poincaré GloVe as initial word embedding, 2-hop computational graph without anchor's siblings, with both relative and absolute position embedding. "i/o" means "instead of", "w/o" means "without".

In lines 8 to 10, we investigate the effect of positional embeddings. A larger performance drop is caused if we remove relative position embeddings (line 8), in comparison to a lesser drop when removing the absolute position embedding (line 9). We hypothesize that the absolute position embedding (depth information) is provided implicitly in the ego graph by edges among events. Line 10

shows that both embeddings are essential to boost the performance by almost 4% gain in recall@10.

5 Related Works

Our work is connected to two lines of research.

Taxonomy Expansion Taxonomy expansion task fits in real-world application scenario that automatically attach new concepts or terms into a human curated seed taxonomy (Vedula et al., 2018). Traditional methods leverage pre-defined patterns to extract hypernym-hyponym pairs for taxonomy expansion (Nakashole et al., 2012; Jiang et al., 2017; Agichtein and Gravano, 2000). Some works use external data and expand taxonomy in a specific domain. For example, Toral et al. (2008) use Wikipedia named entities to expand WordNet, Wang et al. (2014) use query logs to expand search engine category taxonomy. Some works expand a generic taxonomy without using external resources. For example, Shwartz et al. (2016) encode taxonomy traversal paths to seize on the dependency between concepts, Shen et al. (2020) use a GNN model that handles this task, ARBORIST (Manzoor et al., 2020) produces concept representations using signals from both edge semantics and surface forms of concepts. STEAM (Yu et al., 2020) formulates the taxonomy expansion task as a mini-path-based prediction task and introduces a co-training process

for semi-supervised learning. Recently, Zhang et al. (2021) propose the taxonomy completion task in which the new concept can be inserted between existing concepts on taxonomy. Zhang et al. also introduce the TMN model whose auxiliary scorers capture different fine-grained signals. Comparing with these methods using Euclidean space, HYPER-EXPAN uses hyperbolic representation learning to provide feature space with low distortion especially for lower-level concepts on taxonomies.

Hyperbolic Representation Learning Nickel and Kiela (2017) present an efficient algorithm to learn embeddings in a supervised manner based on Riemannian optimization and shows it performs well on link prediction task even with a smaller dimension. Ganea et al. (2018) presents common neural network operations in hyperbolic space and Liu et al. (2019b) extends GNN operations to Riemannian manifolds with differentiable exponential and logarithmic maps. Most related to our work, Chami et al. (2019) derives Graph Convolutional Neural Network (GCN)'s operations in the Lorentz model of hyperbolic space. Hyperbolic representation learning is broadly applied to lexical representations (Dhingra et al., 2018; Tifrea et al., 2019; Zhu et al., 2020), organizational chart induction (Chen and Quirk, 2019), hierarchical classification (López and Strube, 2020; Chen et al., 2020), knowledge association (Sun et al., 2020), knowledge graph completion (Wang et al., 2021a; Balazevic et al., 2019) and event prediction (Surís et al., 2021). A more comprehensive summarization is given in a recent survey by Peng et al. (2021).

There are studies that leverage hyperbolic representation learning to perform taxonomy extraction from text, which are connected to this work. Such studies use Poincaré embeddings trained by hypernymy pairs extracted by lexical-syntactic patterns (Hearst, 1992) to infer missing nodes (Le et al., 2019) and refine preexisting taxonomies (Aly et al., 2019). The patterns suffer from missing and incorrect extractions, and are dedicated to capturing hypernymy relations between nouns. Hence, only terms that are recognizable by the designed patterns are able to be attached to the taxonomy. These works solely rely on graph structures of the taxonomy to obtain hyperbolic embeddings of known concepts, and cannot handle emerging, unseen concepts using their profile information. This is one of the problems that are addressed in this work.

6 Conclusion and Future Work

We present HYPEREXPAN, a taxonomy expansion model which better preserves the taxonomical structure in an expressive hyperbolic space. We use an HGNN to incorporate neighborhood information and positional features of concepts, as well as profile features that are essential to jump-start zero-shot concept representations. Experimental results on WordNet and Microsoft Academic Graph taxonomies show that HYPEREXPAN performs better than its Euclidean counterparts and consistently outperforms state-of-the-art taxonomy expansion models. In the future, we plan to extend HYPEREX-PAN for inducing dynamic taxonomies (Zhu et al., 2021) and taxonomy alignment (Sun et al., 2020).

Ethical Considerations

This work does not present any direct societal consequence. The proposed method aims at improving representation learning to support automated expansion of taxonomies. We believe this study leads to intellectual merits that benefit from automated knowledge acquisition for constructing knowledge representations with complex or sparse structures. It could also potentially lead to broad impacts, as the obtained taxonomical knowledge representations can support various knowledge-driven tasks. It is important to note that the precision of top taxonomy expansion predictions is still not high even with the state-of-the-art method, so human validation is needed before the taxonomy generated by the automated method is used in real-world applications. All experiments are conducted on open datasets.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM conference on Digital libraries*, pages 85–94.

Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4811–4817.

Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. 2020. Constant curvature graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 486–496.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4465–4475.

Eugenio Beltrami. 1868. Teoria fondamentale degli spazii di curvatura costante. *Annali di Matematica Pura ed Applicata (1867-1897)*, 2(1):232–255.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. 1997. Hyperbolic geometry. *Flavors of geometry*, 31:59–115.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4869–4880.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7496–7503.

Muhao Chen and Chris Quirk. 2019. Embedding edge-attributed relational hierarchies. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 873–876.

Michel Marie Deza and Monique Laurent. 2009. *Geometry of cuts and metrics*, volume 15.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 59–69.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5350–5360.

Çaglar Gülçehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter W. Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. Hyperbolic attention networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Mikael Härlin and Per Sundberg. 1998. Taxonomy and philosophy of names. *Biology and Philosophy*, 13(2):233–244.

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

Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2016. Understand short texts by harvesting and analyzing semantic knowledge. *IEEE transactions on Knowledge and data Engineering*, 29(3):499–512.

Birger Iversen and Iversen Birger. 1992. *Hyperbolic geometry*, volume 25.

Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M. Kaplan, Timothy P. Hanratty, and Jiawei Han. 2017. Metapad: Meta pattern discovery from massive text corpora. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 877–886.

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.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. 2020. Geoopt: Riemannian optimization in pytorch.

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3231–3241.

Bang Liu, Weidong Guo, Di Niu, Chaoyue Wang, Shunnan Xu, Jinghong Lin, Kunfeng Lai, and Yu Xu. 2019a. A user-centered concept mining system for query and document understanding at tencent. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1831–1841.

Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019b. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8228–8239.

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards better context-aware lexical semantics:adjusting contextualized representations through static anchors. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4066–4075.

Federico López and Michael Strube. 2020. A fully hyperbolic neural model for hierarchical multi-class classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 460–475.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2044–2054.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv preprint*, abs/1807.03748.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and G. Zhao. 2021. Hyperbolic deep neural networks: A survey. *ArXiv preprint*, abs/2101.04562.

Rik Sarkar. 2011. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer.

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 *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 486–497.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398.

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.

Darin Stewart. 2008. *Building enterprise taxonomies*.

Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. Knowledge association with hyperbolic knowledge graph embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5704–5716.

Dídac Surís, Ruoshi Liu, and Carl Vondrick. 2021. Learning the predictability of the future. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12617.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Antonio Toral, Rafael Muñoz, and Monica Monachini. 2008. Named entity WordNet. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Abraham A Ungar. 2001. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2):135–147.

Nikhita Vedula, Patrick K. Nicholson, Deepak Ajwani, Sourav Dutta, Alessandra Sala, and Srinivasan Parthasarathy. 2018. Enriching taxonomies with functional domain knowledge. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 745–754.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio.

4192

2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203.

Jingjing Wang, Changsung Kang, Yi Chang, and Jiawei Han. 2014. A hierarchical dirichlet model for taxonomy expansion for search engines. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 961–970.

Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. 2021a. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1761–1771.

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

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Carl Yang, Jieyu Zhang, and Jiawei Han. 2020. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 721–730. IEEE.

Shuo Yang, Lei Zou, Zhongyuan Wang, Jun Yan, and Ji-Rong Wen. 2017. Efficiently answering technical questions - A knowledge graph approach. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3111–3118.

Xiaoxin Yin and Sarthak Shah. 2010. Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 1001–1010.

Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7134–7143.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. STEAM: self-supervised taxonomy expansion with mini-paths. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1026–1035.

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.

Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander J. Smola. 2014. Taxonomy discovery for personalized recommendation. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 243–252.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yudong Zhu, Di Zhou, Jinghui Xiao, Xin Jiang, Xiao Chen, and Qun Liu. 2020. HyperText: Endowing FastText with hyperbolic geometry. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1166–1171.

4193

A Graph Convolutional Networks

Graph convolutional network (GCN) (Kipf and Welling, 2017) is a widely-used variant of graph neural network. GCN defines one hop of graph message passing as a combination of the feature transformation and the neighborhood aggregation at a single layer l. The input feature transformation is defined as:

$$\mathbf{h}_i^{\ell,\mathcal{E}} = W^\ell \mathbf{x}_i^{\ell-1,\mathcal{E}} + \mathbf{b}^\ell$$

where $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{E}\}$ is a set of neighboring nodes of node i, W^ℓ and b^ℓ are learnable weight and bias parameters for layer l. The neighborhood aggregation is then defined as:

$$\text{AGG}^0\left(\mathbf{x}^{\ell,\mathcal{E}}\right)_i = \sigma\left(\mathbf{h}_i^{\ell,\mathcal{E}} + \sum_{j \in \mathcal{N}(i)} w_{ij}\mathbf{h}_j^{\ell,\mathcal{E}}\right)$$

where w_{ij} denotes the scores for a weighted aggregation, *i.e.* how important node j is for node i, and σ is a non-linear activation function. By cascading multiple layers of GCN, the message can be propagated over several hops of neighborhoods. The node embeddings in the graph are being updated during the training process. Notice that the superscript 0 in the above equation denotes the 0-curved space, *i.e.* , the aggregation is performed in a Euclidean space.

B Implementation Details

All the models in this work are trained on a single Nvidia A100 GPU[5] on a Ubuntu 20.04.2 operating system. The hyperparameters for each model are manually tuned against different datasets, and the checkpoints used to evaluate are selected by the best performing ones on the development set.

Our entire code-base is implemented in Py-Torch.[6] The implementations of the transformer-based models are extended from the hugging-face[7] code base (Wolf et al., 2020). The implementations of the models compared with, *i.e.* TMN, TaxoExpan and ARBORIST, are obtained and adapted from the original author released code repositories.

B.1 Hyper-parameters

We introduce the hyper-parameters used throughout this work and the searching bounds for the manual hyper-parameter tuning in Table 5.

Type	Batch Size	Initial LR
Bound (lower–upper)	8-128	1×10^{-2}–1×10^{-6}
Number of Trials	2–4	2–3

Table 5: **Search bounds:** for the hyperparameters of all the models.

We set burnin epoch number to 20 during which we use 1e-5 learning rate, after the burnin epochs, the learning rate is 1e-3 with ReduceLROnPlateau scheduler with 10 patience epochs. For each positive sample, we generate 31 negative samples. Dimension for anchor concept representation (output dimension of HGNN) is set to 100. We use two GNN layers by default. We use stochastic Riemannian Adam optimizer (Kochurov et al., 2020; Nickel and Kiela, 2017). For absolute and relative positional embedding, we use 50 dimensions by default. We use MRR of the validation set as the metric to monitor for an early stop.

---

[5]https://www.nvidia.com/en-us/data-center/a100/

[6]https://pytorch.org/

[7]https://github.com/huggingface/transformers