Expanding Taxonomies with Implicit Edge Semantics

Emaad Manzoor*
Carnegie Mellon University
emaad@cmu.edu

Rui Li, Dhananjay Shrouty
Pinterest
{rli,dshrouty}@pinterest.com

Jure Leskovec
Pinterest, Stanford University
jure@{pinterest.com,cs.stanford.edu}

ABSTRACT

Curated taxonomies enhance the performance of machine-learning systems via high-quality structured knowledge. However, manually curating a large and rapidly-evolving taxonomy is infeasible. In this work, we propose Arborist, an approach to automatically expand textual taxonomies by predicting the parents of new taxonomy nodes. Unlike previous work, Arborist handles the more challenging scenario of taxonomies with *heterogeneous* edge semantics that are *unobserved*. Arborist learns latent representations of the edge semantics along with embeddings of the taxonomy nodes to measure taxonomic relatedness between node pairs. Arborist is then trained by optimizing a large-margin ranking loss with a dynamic margin function. We propose a principled formulation of the margin function, which theoretically guarantees that Arborist minimizes an upper-bound on the shortest-path distance between the predicted parents and actual parents in the taxonomy. Via extensive evaluation on a curated taxonomy at Pinterest and several public datasets, we demonstrate that Arborist outperforms the state-of-the-art, achieving up to 59% in mean reciprocal rank and 83% in recall at 15. We also explore the ability of Arborist to infer nodes' *taxonomic-roles*, without explicit supervision on this task.

ACM Reference Format:
Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding Taxonomies with Implicit Edge Semantics. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380271

1 INTRODUCTION

Curated taxonomies have improved the performance of machine-learning systems on a variety of tasks via high-quality structured knowledge. Classification using class-label taxonomies has been shown to be more accurate, both empirically [14, 36, 64, 71] and theoretically [5, 6]. Products, movie and music taxonomies have enabled more relevant recommendations in both the static and sequential settings [25, 27, 28, 79, 81]. Taxonomies have also powered advances in web search [1], user-behavior modeling [38] and model interpretability [34].

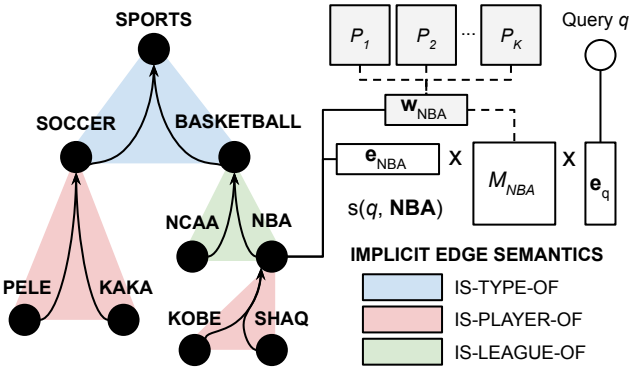


Figure 1: Overview of Arborist expanding a taxonomy with implicit, heterogeneous edge semantics (colored). Arborist learns to measure the taxonomic relatedness $s(q, v)$ of a query-node pair with feature-vectors $\mathbf{e}_q$ and $\mathbf{e}_v$. Relatedness is defined via linear maps $P_1, \ldots, P_K$ and node-embeddings $\mathbf{w}_v$, jointly trained to minimize a large-margin ranking loss.

Several platforms[1–3] have come to rely on curated taxonomies to power their user-profiling, content categorization and recommendation systems. With the influx of new content on the platform, it is crucial to update these taxonomies to remain relevant. However, manual curation in the face of rapidly-evolving content is infeasible. We thus focus on the problem of automatic taxonomy expansion. We are specifically motivated by the task of expanding the *Pinterest Taxonomy*: a curated hierarchy of over 10,000 *interests* that forms the backbone of user and content understanding at Pinterest. Interests are textual phrases describing concepts that may be general (such as *architecture* and *health*) or specific (such as *mid-century architecture* and *mental well-being*). Interests are used to categorize both Pinterest users and the content they create, and subsequently used for recommendations [17, 33] and interest-based ad targeting.

In the most recent taxonomy expansion effort, 8 curators appended a total of 6,000 new taxonomy nodes in a month [23]: a total of over 500 taxonomist-hours (by conservative estimates) at the rate of 5 minutes per new taxonomy node. This is in stark contrast to the millions of new visual bookmarks added by Pinterest users every day day, each potentially introducing a new interest to the platform. Automatic taxonomy expansion is thus the only feasible approach to maintain the relevance of recommended content and ads as the platform experiences rapid growth.

The taxonomy expansion problem is challenging for several reasons. For large-scale taxonomies having thousands of nodes, any query node (to be appended to the taxonomy) has a plethora of potential parents. Learning a binary classifier to predict whether a query-node pair is taxonomically related is afflicted by the issue

*Work done while the author was an intern at Pinterest.
[1]Google Shopping: http://google.com/basepages/producttype/taxonomy.en-US.txt
[2]Mozilla Website Directory: http://dmoztools.net
[3]The Pinterest Taxonomy: [23]

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380271

of lexical memorization [30]. Instead of learning to represent taxonomic relationships, simple classifiers learn that certain nodes are *prototypical parents*. For any query-node pair where the node is a prototypical parent, the classifiers (often incorrectly) predict the existence of a taxonomic relationship.

Hence, it is crucial to explicitly model taxonomic relationships. This is difficult when the taxonomy has *heterogeneous edge semantics*. For example, the Pinterest Taxonomy contains both IS-IN edges connecting places and IS-TYPE-OF edges connecting dog breeds. Moreover, the edge semantics are not explicitly recorded during the taxonomic curation process: they are *implicit and unobserved*. In addition, the business-critical nature of curated taxonomies mandates that automatic taxonomy expansion be overseen by in-house experts, or risk causing business-wide failures. This introduces the additional burden of *easy human-verification*.

Recent successes in textual taxonomy expansion were based on learning transformations of word-embeddings [18, 20, 62, 75] and comprise the current state-of-the-art [9, 12]. However, these approaches assume homogeneous edge semantics. Knowledge-base embeddings [11, 46, 61] and hierarchical embeddings [21, 44, 63] enable accurate link-prediction in knowledge graphs with heterogeneous, explicit edge-types. Yet, neither embedding approach can handle unobserved edge-types, or construct embeddings for unseen nodes. Finally, the issue of ease of human-verification in automatic taxonomy expansion remains unexplored.

We thus propose ARBORIST, a method to expand textual taxonomies with implicit edge semantics. ARBORIST learns node embeddings, along with linear maps that are shared across the taxonomy, to represent the latent edge semantics. The combination of local and global taxonomic information equips ARBORIST with the flexibility to capture heterogeneous edge semantics while remaining robust to missing or misplaced nodes. The node embeddings and linear maps are trained to optimize a taxonomic-relatedness score, such that child-parent node pairs are scored higher than other pairs by a large margin. Distinct from previous approaches, ARBORIST is guaranteed to predict parents that are near the actual parents in the taxonomy: this facilitates human-verification by restricting corrections to small taxonomic neighborhoods. Fig. 1 illustrates ARBORIST operating on the Pinterest Taxonomy.

In summary, we make the following contributions:

I. We introduce ARBORIST, a taxonomy expansion method that combines global and local taxonomic information to explicitly model heterogeneous and unobserved edge semantics.
II. We prove that ARBORIST minimizes an upper-bound on the shortest-path distances between the predicted and actual taxonomy parents. This property is invaluable for human verification, since incorrectly appended nodes may be easily corrected by probing a small local taxonomy neighborhood.
III. We evaluate ARBORIST on several datasets, metrics and compare to several baselines to demonstrate performance exceeding the present state-of-the-art. We conduct an in-depth ablation study to analyze the effect of each hyperparameter.
IV. We explore the ability of ARBORIST to infer nodes' *taxonomic-roles*, without being explicitly trained for role-classification.

Reproducibility: Code and data (for the public taxonomies) are available at http://cmuarborist.github.io.

	Edge Semantics		*Predictions*
Method	Heterogeneous	Implicit	Inductive
Hypernym Prediction	✗	✗	✓
via Lexical Patterns [26, 42, 57]			
via Word Embeddings [8, 51, 56, 67, 70]			
via Projection Learning [9, 20, 62, 75]			
Knowledge-Graph Embeddings [11, 46, 61]	✓	✗	✗
Hierarchical Embeddings	✓	✓	✗
in Euclidean Space [15, 31, 43, 63, 65]			
in Hyperbolic Space [3, 13, 21, 44]			
in Gaussian Space [66]			
HIEXPAN [55]	✗	✓	✓
ARBORIST (this work)	✓	✓	✓

Table 1: Related work. Inductive approaches can add both new nodes and new edges. HIEXPAN and lexical-pattern hypernym predictors require a text corpus.

2 RELATED WORK

Taxonomy expansion is related to hypernym prediction. Hypernym predictors learn to represent the IS-A relationship using a training taxonomy with homogeneous IS-A semantics (such as WordNet [40]) and (optionally) a textual corpus. While earlier approaches relied on lexical patterns [26, 42, 57], recent methods learn classifiers operating on pre-trained word-embeddings [8, 51, 56, 67, 70]. A few methods learn word-embeddings tailored for hypernym prediction [4, 77], but cannot infer the embeddings of unseen nodes to be added to the taxonomy (they are not inductive). Taxonomy learning via hypernym prediction is surveyed in [68].

The state-of-the-art in hypernym prediction is via piecewise projection-learning on word-embeddings [20]. In this approach, a projection matrix is learned for each cluster of the word-embedding space to minimize the L_2-norm between the projected child's word-embedding and parent's word-embedding. This approach was later extended to use randomly sampled non-parents (negative samples) during training [62] and to jointly learn the clusters and their projection matrices [75]. A variant of projection-learning called CRIM [9] incorporates negative sampling, word-embedding fine-tuning and multiple projection matrices to achieve the best performance in the SEMEVAL 2018 hypernym discovery task [12]. We compare ARBORIST with CRIM in our evaluation.

HIEXPAN [55] expands taxonomies with unobserved but homogeneous edge semantics, given a training taxonomy and textual corpus. The method iteratively expands the taxonomy-width via set-expansion and taxonomy-depth via weakly-supervised relation extraction. Both subprocedures require a large textual corpus, which is unavailable in our scenario. The expansion is task-guided, since the edge semantics are inferred from the training taxonomy instead of being explicitly provided.

Knowledge-base completion is the closely related task of inferring missing links in knowledge-graphs with heterogeneous edge semantics that are explicitly observed. Recent approaches learn embeddings for nodes and edge types and predict links via algebraic operations [11, 46, 58, 61], which is reminiscent of earlier work on learning relational embeddings [47, 53, 54]. Since the embeddings are derived from the given knowledge-graph, they cannot be inferred for unseen nodes or unobserved edge types.

Hierarchical embeddings have been recently proposed that learn embeddings for taxonomy nodes in Euclidean [15, 31, 43, 63, 65], hyperbolic [3, 13, 21, 44, 45] or other expressive spaces [66]. These embeddings may be supervised by a training taxonomy, or unsupervised and operate on a textual corpus. Hierarchical embeddings do not assume specific lexical relationship types or homogeneity in the taxonomy. However, current hierarchical embedding methods cannot infer the embeddings of unseen taxonomy nodes, and are only effective in predicting missing taxonomy links.

Taxonomy induction is the related unsupervised task of constructing a taxonomy from a textual corpus [7, 35, 76]. Though the target taxonomy semantics are typically fixed to known lexical relationships (such as IS-A or IS-IN), recent work has extended taxonomy induction to accommodate arbitrary edge semantics [78]. Taxonomy expansion is complementary to taxonomy induction, and is the logical next step in the taxonomy learning process [68].

Our proposed method draws on several theoretical concepts from similarity and distance metric-learning [41, 52, 74]; specifically their large-margin [71, 72], hierarchical [14, 64] and learning-to-rank [32, 37] formulations. We are also inspired by work on graph-embeddings [24, 49, 60]; specifically on structural-role embeddings [2, 16, 50]. With our development of ARBORIST we establish principled connections between these learning tasks and the taxonomy expansion problem.

3 PROBLEM OVERVIEW

We now formalize the taxonomy expansion problem. Denote by $\mathcal{T} = (V, E)$ the taxonomy to be expanded, represented as a directed acyclic graph. Each node $u \in V$ is a textual phrase and each directed edge $(u, v) \in E$ represents a taxonomic relationship between a child $u \in V$ and its parent $v \in V$. We assume that each node u is equipped with a feature-vector $\mathbf{e}_u \in \mathbb{R}^d$ derived using an external procedure, such as via unsupervised word-embeddings [39, 48, 59].

We impose weak assumptions on the nature of the taxonomy. Specifically, we expect child-parent relationships in the taxonomy to satisfy the distributional inclusion hypothesis [22]: the parent is (i) related to, and (ii) more general than the child. We also expect the taxonomy to follow a hierarchical structure: each child is only linked to its least-general related nodes. Note that a taxonomy node may have multiple parents.

The specific semantics of the edges are unobserved and possibly heterogeneous, depending on the taxonomist's business goals. For example, geographic taxonomies have homogeneous edge semantics (such as IS-IN) whereas more general taxonomies such as the Pinterest Taxonomy exhibit heterogeneous edge semantics (such as IS-SUBCLASS-OF and IS-PART-OF).

Given an unseen *query node* and a ground-truth taxonomy, our goal is to append the query node to the taxonomy as a new leaf node. In practice, in-house taxonomists expect an ordered list of predicted parents for the query node that can be verified before updating the taxonomy. Thus, we formalize this objective as the following ranking problem:

PROBLEM 1 (TAXONOMY EXPANSION). *Given a taxonomy $\mathcal{T} = (V, E)$, feature vectors $\mathbf{e}_u \in \mathbb{R}^d$ for each node $u \in V$ and a query $q \notin V$ with feature vector $\mathbf{e}_q \in \mathbb{R}^d$, rank the taxonomy nodes such that the true parent(s) of q are ranked higher than its non-parents.*

4 PROPOSED METHOD: ARBORIST

At its core, ARBORIST learns to measure the taxonomic relatedness of node-pairs as a function of their respective feature vectors (§4.1). Taxonomic relatedness is defined in terms of two learnable parameters: (i) embeddings of the taxonomy nodes capturing their *taxonomic roles*, and (ii) a collection of linear maps that are shared across the taxonomy. The parameters are trained to minimize a large-margin ranking loss (§4.2), with a theoretically-grounded dynamic margin function (§4.3). To ensure fast convergence, ARBORIST employs distance-weighted negative sampling (§4.4). We now describe ARBORIST in detail.

4.1 Measuring Taxonomic Relatedness

We seek to define a function $s : V \times V \rightarrow \mathbb{R}$ that can be trained to measure the taxonomic relatedness of node-pairs. For a child node u and parent node v, a straightforward definition of s using a linear map $M \in \mathbb{R}^{d \times d}$ is as follows:

$$s(u, v) = (\mathbf{e}_u M) \cdot \mathbf{e}_v. \tag{1}$$

where M is learned from the child-parent node-pairs in the ground-truth taxonomy. Intuitively, M represents each taxonomic relationship as a linear transformation in the nodes' feature-space. However, heterogeneous edge semantics may not be represented well by a single transformation. Hence, we capture the heterogeneity in edge semantics using a different linear map M_v at each parent node v:

$$s(u, v) = (\mathbf{e}_u M_v) \cdot \mathbf{e}_v. \tag{2}$$

A similar formulation was used in learning hierarchical similarity metrics [64]. A critical problem with this formulation is the explosion in the number of parameters to $O(d^2|V|)$, where typically $d = O(100)$ and $|V| = O(1000)$. Additionally, the training data is fragmented into node-level subsets to train each linear map. Overall, this constrains the method's scalability, reduces its robustness to training data with missing or misplaced taxonomy nodes, and increases its tendency to overfit.

We mitigate this problem by exploiting the fact that the number of distinct edge semantics in a taxonomy is often much smaller than $O(|V|)$. Concretely, we assume k latent edge semantics and define each linear map as a weighted combination of k linear-maps $P_1, \ldots, P_k \in \mathbb{R}^{d \times d}$ that are shared across all taxonomy nodes:

$$M_v = \sum_{i=1}^{k} \mathbf{w}_v[i] \times P_i. \tag{3}$$

Here, $\mathbf{w}_v \in \mathbb{R}^k$ is an embedding of the node v to be learned. This formulation reduces the number of parameters to $O(d^2k + k|V|)$ without fragmenting the training data.

We further constrain the node-embeddings by defining them in terms of the node feature-vectors as follows:

$$\mathbf{w}_v = f(\mathbf{e}_\mathbf{v}) \tag{4}$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ is an arbitrary function to be learned. Constraining the node-embeddings in this manner improves robustness for nodes having a large number of missing children, which are more susceptible to noise. We instantiate f as a simple linear-map; we did not see significant gains from more complex functions.

4.2 Learning and Prediction

We now seek to rank the taxonomy nodes for a given query node such that its true parents are ranked higher than other nodes. To improve robustness and generalization, we train the model such that child-parent node-pairs have higher taxonomic relatedness than other node-pairs by a large margin. Formally, our goal is to satisfy the following constraint for every child-parent pair $(u, v) \in E$:

$$s(u, v) \geq s(u, v') + \gamma(u, v, v') \quad \forall v' \in V - H(u), \tag{5}$$

where $H(u) \subset V$ is the set of parents of u, v' is a non-parent node and $\gamma(u, v, v')$ is the desired margin defined as a function of the child, parent and non-parent nodes.

We now derive the loss function to be minimized in order to satisfy the large-margin constraint (5). Denote by $\mathcal{E}(u, v, v')$ the degree to which a non-parent node v' violates the large-margin constraint of child-parent pair (u, v):

$$\mathcal{E}(u, v, v') = \max[0, s(u, v') - s(u, v) + \gamma(u, v, v')]. \tag{6}$$

When the large-margin constraint is satisfied, $\mathcal{E}(u, v, v') = 0$ and the non-parent incurs no violation. Otherwise, $\mathcal{E}(u, v, v') > 0$.

The overall loss function $\mathcal{L}(\mathcal{T})$ is the total violation of the large-margin constraints by the non-parents corresponding to every child-parent pair (u, v):

$$\mathcal{L}(\mathcal{T}) = \sum_{(u,v)\in E} \sum_{v' \in V - H(u)} \mathcal{E}(u, v, v') \tag{7}$$

The node embeddings $\mathbf{w}_v$ and linear-maps $P_1, \ldots, P_k$ are jointly trained to minimize $\mathcal{L}(\mathcal{T})$ via gradient-descent. Given the trained parameters and a query node $q \notin V$ having feature-vector $\mathbf{e}_q$, predictions are made by ranking the taxonomy nodes v in decreasing order of their taxonomic relatedness $s(q, v)$.

4.3 Dynamic Margin Function

An L_2-norm variant of the large-margin loss in (7) was recently proposed as the *triplet loss* [52] for similarity-learning in computer vision. The triplet loss employs a constant margin γ for all training pairs, which is tuned via cross-validation. To accommodate more complex decision boundaries, several approaches extend the triplet loss with dynamic or adaptive margins, set using heuristics [69] or learned from the data [19].

We propose a principled dynamic margin function that requires no tuning, learning or heuristics. We relate the margin to the shortest-path distances in the taxonomy between the predicted and true parent nodes. Denote by $d(\cdot, \cdot)$ the undirected shortest-path distance between two nodes in the taxonomy. With the following theorem, we bound the undirected shortest-path distance between the highest-ranked predicted parent $\hat{v}(u) = \arg\max_{\tilde{v}} s(u, \tilde{v})$ and any true parent v for every child node u:

Proposition 1. *When $\gamma(u, v, v') = d(v, v')$, $\mathcal{L}(\mathcal{T})$ is an upper-bound on the sum of the undirected shortest-path distances between the highest-ranked predicted parents and true parents:*

$$\sum_{(u,v)\in E} d(v, \hat{v}(u)) \leq \mathcal{L}(\mathcal{T}).$$

Proof. The highest-ranked predicted parent for a node is its most-related taxonomy node. The following inequality thus follows:

$$s(u, \hat{v}(u)) - s(u, \tilde{v}) \geq 0 \quad \forall \tilde{v} \in V. \tag{8}$$

Since $\gamma(u, v, \hat{v}(u)) = d(v, \hat{v}(u))$, we can use (6) and (8) to lower-bound the large-margin violation of the predicted parent:

$$\begin{aligned} \mathcal{E}(u, v, \hat{v}(u)) &= \max[0, s(u, \hat{v}(u)) - s(u, v) + \gamma(u, v, \hat{v}(u))] \\ &\geq s(u, \hat{v}(u)) - s(u, v) + \gamma(u, v, \hat{v}(u)) \\ &\geq \gamma(u, v, \hat{v}(u)) = d(v, \hat{v}(u)) \end{aligned} \tag{9}$$

Using the fact that $\mathcal{E}(u, v, v') \geq 0$ and summing over child-parent pairs and their corresponding non-parents concludes our proof:

$$\begin{aligned} \mathcal{L}(\mathcal{T}) &= \sum_{(u,v)\in E} \sum_{v' \in V - H(u)} \mathcal{E}(u, v, v') && (10) \\ &\geq \sum_{(u,v)\in E} \mathcal{E}(u, v, \hat{v}(u)) \\ &\geq \sum_{(u,v)\in E} d(v, \hat{v}(u)). \quad \square && (11) \end{aligned}$$

Thus, minimizing the loss in (7) also minimizes an upper-bound on the sum of shortest-path distances between the highest-ranked predictions and true parent nodes. This guarantees that the predicted parent nodes are near the true parent nodes in the ground-truth taxonomy. Intuitively, setting $\gamma(u, v, v') = d(v, v')$ encourages non-parent nodes that are further away in the taxonomy to be scored relatively lower on taxonomic relatedness.

This guarantee is important from the perspective of taxonomic experts; if Arborist incorrectly predicts the parent of a query-node, the taxonomic expert need only probe a small neighborhood around the predicted parent node in the taxonomy in order to find the correct parent. This requires substantially less effort than searching the entire taxonomy for the correct parent.

4.4 Scalable Negative Sampling

Summing over all the non-parents $v' \in V - H(u)$ for each pair $(u, v) \in E$ requires computing $O(VE)$ taxonomic relatedness scores in every iteration of the minimization process, which is infeasible for large-scale taxonomies. To improve scalability, we instead sample m non-parents (or negative samples) from $V - H(u)$.

The negative-sampling distribution plays an important role in the convergence of gradient-based minimization of the large-margin loss. In practice, sampling negatives uniformly results in extremely slow convergence. This is because negative samples that incur zero violation of the large-margin constraint in (6) contribute nothing to the gradient. Hence, heuristics such as "semi-hard negative mining" [52] have been proposed to select more effective negative samples.

We employ a negative-sampling distribution based on distance-weighted negative-sampling [73], which we found provides a favorable tradeoff between the computational effort of negative-sampling and the rate of convergence of minimization. We sample each non-parent v' for a node-pair (u, v) with probability:

$$\Pr(v'|(u, v)) \propto \mathbf{e}_u \cdot \mathbf{e}_{v'}. \tag{12}$$

Dataset	Pinterest	SemEval	Mammal
Number of edges	10,768	18,827	5,765
Number of nodes	10,792	8,154	5,080
Training nodes	7,919	7,374	4543
Test nodes	2,873	780	537
Depth	7	∞	18
Heterogenous Edges	✓	✗	✓

Table 2: Taxonomy datasets. Test sets have 15% of the leaf nodes and their outgoing edges. Taxonomy depth is the longest shortest-path length, or ∞ if directed-cycles exist.

5 EVALUATION

5.1 Datasets and Metrics

We evaluate Arborist in three different scenarios, each corresponding to a different taxonomy dataset (summarized in Table 2).

Semi-synthetic heterogeneous taxonomy (Mammal). We construct a taxonomy from WordNet [40] by extracting the subgraph rooted at `mammal.n.01`, restricted to noun nodes and edges of three types: (i) hypernyms capturing is-a semantics (for example, *rhino* is-a *odd-toed ungulate*), (ii) part-holonyms capturing is-part-of-whole semantics (for example, *hoof* is-part-of-whole *ungulate*), and (iii) substance-holonyms, capturing is-part-of-substance semantics (for example, *collagen* is-part-of-substance *cartilage*). Other edge types in WordNet were either the reverse-form of the selected edge types, or cannot exist in a valid hierarchy (such as the synonym edge type). We compute the 300-dimensional FastText embedding [10] for each node.

Benchmark homogeneous taxonomy (SemEval). We combine the train and test subsets of the *SemEval 1A* taxonomy used in the SemEval 2018 hypernym discovery shared task [12]. While this taxonomy exhibits homogeneous is-a semantics, it is a manually-verified benchmark for taxonomy expansion that enables comparison with the state-of-the-art. As with Mammal, we compute the 300-dimensional FastText embedding [10] for each node.

Gold heterogeneous taxonomy (Pinterest). Pinterest is an expert-curated taxonomy that forms a core component of user and content understanding at Pinterest. The taxonomy nodes may be concrete entities (such as *New York*) or abstract concepts (such as *Mental Wellbeing*) from a wide array of domains such as fashion, health and travel. Each node represents an *interest* associated with a user or visual bookmark on Pinterest. The taxonomy exhibits implicit and heterogeneous edge semantics. We compute the 300-dimensional PinText embedding [80] for each node.

To compute the embeddings of multi-word nodes in any dataset, we mean-pool the embeddings of their constituent words. For each dataset, we hold-out a 15% sample of the leaf nodes as a test subset and use the remainder for training and validation.

Metrics. We measure the ranking quality of the predicted parents using the mean reciprocal rank (MRR). For a test query-parent pair, the reciprocal rank of the true test parent is the multiplicative inverse of its rank in the predicted parents list. If a test query has multiple parents, we use the the reciprocal of the highest ranked true parent. The MRR is the mean of the reciprocal ranks over all test queries, ranging from 0% (worst) to 100% (best).

	Pinterest		SemEval		Mammal	
Method	Acc.(%)	F1 (%)	Acc. (%)	F1 (%)	Acc. (%)	F1 (%)
Vec-Concat	86.216	86.462	62.618	59.255	72.466	72.089
Vec-Sum	88.148	87.741	64.950	60.600	78.294	77.236
Vec-Diff	87.783	87.005	66.469	63.379	77.787	75.716
Vec-Prod	87.121	85.953	68.374	65.670	80.152	78.017

Table 3: Accuracy (Acc.) and F1-score (F1) in percentage of hypernym detectors on binary classification of child-parent pairs. Threshold set to 0.5. For both metrics, 100.0 is best.

We also measure the recall at 15 (Recall@15) of the predicted parents, which is the fraction of test queries for which any of the true test parents lie in the top 15 predicted parents. This metric ignores the actual position of the parent in the predicted parent list.

Finally, we measure the undirected shortest-path distance (SPDist) in the taxonomy between the top-ranked predicted parent and the true test-parent, averaged over all test queries. If the top-ranked predicted parent and the true test-parent lie in disconnected components of the taxonomy, their distance is set to the maximum distance between any two nodes in the taxonomy. This metric quantifies how easily a taxonomic expert can verify and fix incorrectly attached nodes by probing a small neighborhood around their predicted parents.

5.2 Evaluating Hypernym Detectors for Taxonomy Expansion

A straightforward approach to expand taxonomies is via state-of-the-art hypernym detectors [8, 51, 56, 70]. Hypernym detectors are binary classifiers that predict whether a node-pair is taxonomically-related. They operate on features derived from the feature-vectors of the node-pair via simple algebraic operations. We evaluate hypernym detectors based on vector concatenation (VEC-CONCAT), addition (VEC-SUM), subtraction (VEC-DIFF) and elementwise-product (VEC-PROD). For each hypernym detector, we train a binary Random Forest classifier with 100 trees[1] on a balanced training set, constructed by randomly sampling an unconnected node-pair for each connected node-pair in the training data.

We verify that the trained hypernym detectors perform well in classifying taxonomically-related node-pairs, achieving F1 scores between 54-88% (Table 3). Notice that all hypernym detectors perform relatively better on Pinterest than on SemEval and Mammal; a trend we will observe throughout our evaluation. This may be attributed to the feature-vectors for Pinterest being fine-tuned on Pinterest content, in contrast with the feature-vectors for SemEval and Mammal trained on an unrelated web corpus.

Also notice that no single hypernym detector outperforms all the others across datasets. This suggests that taxonomic relationships are difficult to represent universally with simple algebraic operations. Further, the best performing hypernym detector for each dataset achieves the lowest mean shortest-path distance (SPDist) between the predicted and true parents. This empirically motivates the utility of our theoretical guarantee (§4.3) for taxonomy expansion performance.

[1] All other options were set to the `scikit-learn-0.21.3` defaults.

	Pinterest			SemEval			Mammal		
	MRR (%)	Recall@15 (%)	SPDist	MRR (%)	Recall@15 (%)	SPDist	MRR (%)	Recall@15 (%)	SPDist
Vec-Concat	41.831	64.671	3.816	20.992	33.155	3.474	14.995	30.726	4.274
Vec-Sum	33.891	62.548	4.124	17.803	27.607	4.047	19.611	38.175	4.186
Vec-Diff	41.185	67.699	3.494	18.514	30.949	4.163	31.386	46.182	3.674
Vec-Prod	42.233	68.743	3.144	17.483	31.083	4.178	**32.177**	48.976	3.665
CRIM	53.223	79.325	2.393	41.691	62.064	**2.743**	21.345	52.700	4.080
Arborist	**59.044**	**83.606**	**2.220**	**43.373**	**67.694**	2.864	29.354	**61.639**	**3.225**

Table 4: Mean reciprocal rank (MRR) and recall at 15 in percentage, and the mean shortest-path distance (SPDist) between the true parents and the highest-ranked predicted parents. For MRR and recall at 15, 100.0 is best. For SPDist, 0.0 is best. The best performing method for each dataset and metric combination is emphasized in bold.

Query	Predicted Parents
Accurate Predictions (true parents in the top 4 predicted parents)	
luxor	**africa travel**, european travel, asia travel, greece
2nd month baby	**baby stage**, baby, baby names, preparing for baby
depression	mental illness, **stress**, mental wellbeing, disease
ramadan	hosting occasions, **holiday**, sukkot, middle east and african cuisine
minion humor	humor, people humor, **character humor**, funny
Inaccurate Predictions (true parents not in the top 4 predicted parents)	
artificial flowers	planting, dried flowers, DIY flowers, edible seeds
thor	adventure movie, action movie, science movie, adventure games
smartwatch	wearable devices, phone accessories, electronics, computer
disney makeup	halloween makeup, makeup, costume makeup, character makeup
holocaust	history, german history, american history, world war
Predictions for Non-Taxonomic Pinterest Search Queries	
what causes blackheads	skin concern, mental illness, feelings, disease
meatloaf cupcakes	cupcakes, desserts, no bake meals, steak
benefits of raw carrots	food and drinks, vegetables, diet, healthy recipes
kids alarm clock	toddlers and preschoolers, child care, baby sleep issues, baby
humorus texts	poems, quotes, authors, religious studies

Table 5: Top 4 predicted parents by Arborist on Pinterest for a sample of test queries (top, middle) and search queries not present in the taxonomy (bottom). The true parent for each query is emphasized in bold.

We now evaluate hypernym detectors on the taxonomy expansion task. We use the predicted probability of a node-pair being taxonomically related to rank the taxonomy nodes for each test query node. The results are in Table 4. The results demonstrate that hypernym detectors can serve as strong baselines for taxonomy expansion. This is especially true for high-quality feature-vectors like PinText, which may capture hierarchical relationships between taxonomy nodes by virtue of being fine-tuned on relevant corpora.

Intriguingly, hypernym detectors outperform all other methods on Mammal (in terms of the MRR). The nodes in the Mammal taxonomy are dominated by scientific terms, which are rare in the crawled web corpora used to train the input FastText embeddings. Hence, the embeddings may not contain enough information to infer hierarchical relationships. With such information-poor embeddings (with respect to the given taxonomy), simpler models such as hypernym detectors may be preferable. Hypernym detectors are worse off on other metrics, but not by a significant margin.

5.3 Comparing Arborist with k-Projection Learning (CRIM)

We now compare Arborist with CRIM [9], a variant of projection-learning [20] that outperformed competing methods by a significant margin on the SemEval 2018 hypernym discovery shared task [12]. CRIM learns a global weighted combination M of k linear-maps to represent all taxonomic relationships. The taxonomic relatedness of a node-pair u, v is measured as $s(u, v) = (\mathbf{e}_u M) \cdot \mathbf{e}_v$, where $\mathbf{e}_u$ and $\mathbf{e}_v$ are the node-pair feature vectors. M is trained to minimize a binary cross-entropy loss similar to that of word2vec [39].

We reimplement the supervised (corpus-free) variant of CRIM with the initialization, fine-tuning and negative-sampling heuristics described in [12]. We discard the positive-subsampling and multi-task learning heuristics that were found to be detrimental. We optimize both Arborist and CRIM using Adam [29] and tune their hyperparameters on a validation subset of the training data. Since Arborist and CRIM converge at different rates, we train both

methods for 150 epochs (for Pinterest and SemEval) or 500 epochs (for Mammal) and select the trained model at the epoch with the highest validation MRR (see appendix for details).

Taxonomy expansion results are reported in Table 4. Overall, Arborist and CRIM improve over the hypernym detectors on all datasets and evaluation metrics, by over 200% in some cases. This justifies explicitly optimizing for the taxonomy expansion ranking task, and representing taxonomic relationships with more complex functions of the node-pair feature-vectors. Arborist outperforms CRIM on all datasets and evaluation metrics. Notably, Arborist gracefully degrades to similar performance as CRIM on the SemEval taxonomy with *homogeneous* edge semantics.

Table 5 reports the top-ranked predicted parents by Arborist on Pinterest for both accurately and inaccurately-predicted test queries (true parents are emphasized in bold). The results showcase predictions on a variety of node-types present in the Pinterest taxonomy, from concrete entities such as locations (*Luxor*) and fictional characters (*Thor*) to abstract concepts such as *depression*. We observe that even inaccurately-predicted parents conform to some notion of relatedness and immediate hierarchy, suggesting potentially missing edges in the taxonomy.

We also showcase Arborist's predictions for *search queries* made on Pinterest that are *not present* in the taxonomy (Table 5, bottom). Qualitatively, Arborist is able to accurately associate unseen natural language queries to potentially related nodes in the Pinterest taxonomy. Of note is the search query *what causes blackheads*, which is not just associated with its obvious parent *skin concern*, but also to the very relevant parent *feelings*.

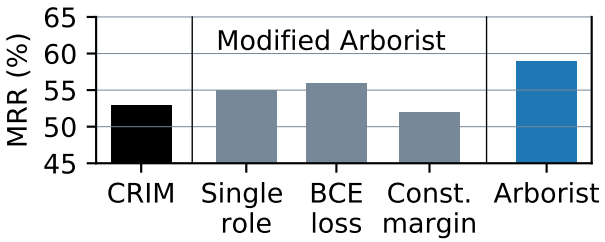
5.4 Ablation Study

The performance of Arborist may be attributed to two key modeling choices: (i) learning node-specific embeddings $\mathbf{w}_v$ to capture heterogeneous edge semantics, and (ii) optimizing a large-margin ranking loss with dynamic margins. We now evaluate the impact of each of these modeling choices on taxonomy expansion performance with a suite of ablation experiments conducted on Pinterest. We use CRIM as a baseline. The results are summarized in Fig. 2.

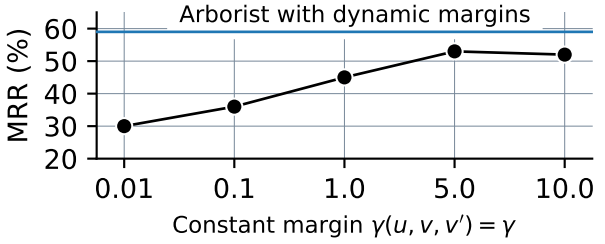
We first restrict Arborist to a single embedding for all nodes (termed *single-role*). With this restriction, Arborist can no longer capture heterogeneous edge semantics. Performance gains over CRIM can only attributed to the large-margin loss. Fig. 2(a) shows that Arborist still improves over CRIM by 2 percentage points, demonstrating the utility of the large-margin loss.

Next, we allow Arborist to capture heterogeneous edge semantics, but modify it to minimize the BCE loss used by CRIM. In this configuration, performance gains over CRIM can be attributed to capturing heterogeneous edge semantics. From Fig. 2(a), we find that this configuration improves over CRIM by 4 percentage points. Thus, capturing heterogeneous edge semantics has a greater marginal impact on performance than the choice of loss function. Given the technical difficulty of optimizing large-margin losses (which converge slowly and require long training times), Arborist with the BCE loss is a pragmatic short-term alternative.

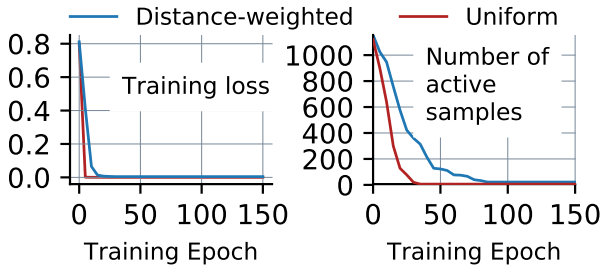
Finally, we modify Arborist to use constant margins and evaluate its performance with different margin values. The performance at the best margin value is reported in Fig. 2(a). We find that the



(a) Summary of ablation study



(b) MRR for each value of constant margin vs. dynamic margins



(c) Uniform vs. distance-weighted negative-sampling

Figure 2: Ablation study of Arborist on Pinterest: (a) summary of results, (b) effect of dynamic margins vs. constant margins, (c) effect of uniform negative sampling vs. distance-weighted negative sampling.

restriction to constant margins affects performance severely, in addition to introducing another hyperparameter to tune. For no constant margin-value did performance exceed that of Arborist with dynamic margins (Fig. 2(b)).

The negative-sampling distribution plays a critical role in the convergence of the loss function. To understand its effect on taxonomy expansion performance, Fig. 2(c) (left) shows the training loss of Arborist at each training epoch with uniform negative sampling and distance-weighted negative sampling. With uniform negative-sampling, the training loss quickly drops to near-zero, leading to extremely slow convergence. This is due to the lack of *active* negative-samples that contribute non-zero values to the training-loss. This is evident from Fig. 2(c) (right), which shows the number of active negative-samples at each training epoch for uniform and distance-weighted negative sampling. With distance-weighted negative sampling, the number of active samples drops at a slower rate. This prevents the training loss from dropping to near-zero too quickly and facilitates faster convergence.

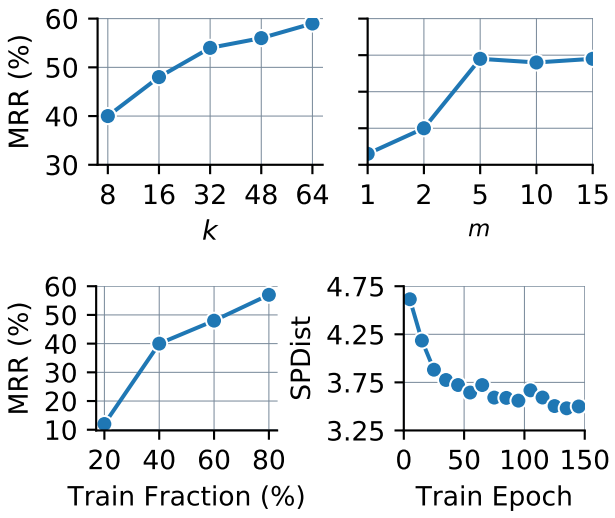


Figure 3: Effect of the number of linear maps k (top-left), the number of negative samples m (top-right) and the training data fraction (bottom-left) on the MRR of Arborist on Pinterest. Also shown (bottom-right) is the average undirected shortest-path distance between predicted and true *test* parents (SPDist) with training epoch.

5.5 Effect of Hyperparameters

Fig. 3 shows the MRR of Arborist on Pinterest for different values of the number of linear maps k (top-left) and the number of negative samples m (top-right). We find that increasing k improves taxonomy expansion performance monotonically, due to the increased model flexibility. Increasing m improves performance to a certain limit, after which there is no significant improvement. A larger m promotes a larger number of *active* negative samples that contribute to the gradient of the loss, leading to faster convergence.

The obtained MRR for different k and m in Fig. 3 also indicates that the performance of Arborist is not sensitive to small variations in k and m. Thus, tuning k and m by cross-validating over a coarse grid of values suffices in practice. Our recommendation is to set k to be as large as possible, within the constraints of the available compute time and memory.

We also evaluate the robustness of Arborist as the fraction of training data available decreases in Fig. 3 (bottom-left). We find that Arborist remains performant until more than half the training data is discarded, at which point it performs on par with hypernym detectors trained on 85% of the taxonomy. This robustness may be attributed to the sharing of information between different nodes via the global linear maps; nodes that lack sufficient incoming edges to learn from may benefit from similar nodes that have sufficient data.

In Fig. 3 (bottom-right), we also demonstrate that Arborist reduces the mean undirected shortest-path distance between the predicted and true *test* parent nodes (SPDist) in each training epoch. While our proposition in §4.3 guarantees the same for the training data, Fig. 3 suggests that the same result may also hold for unseen test data. We delegate formal treatment of this generalization ability to future work.

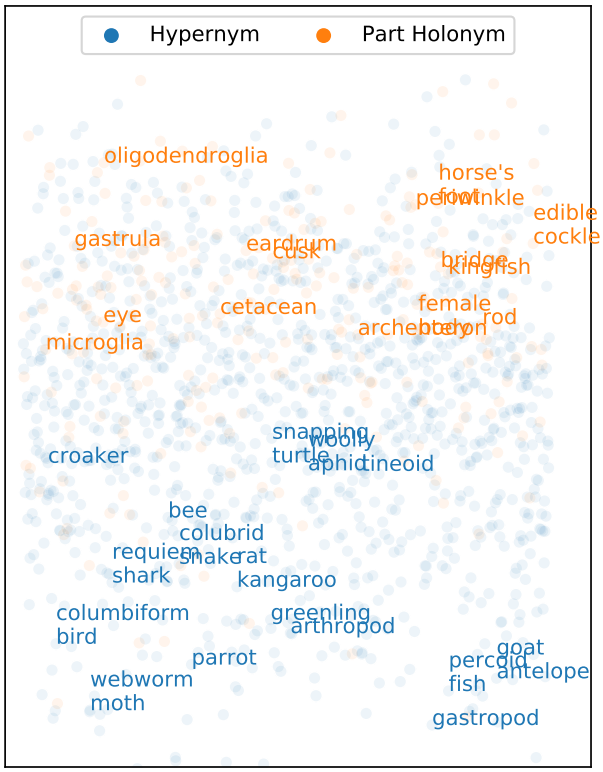


Figure 4: Taxonomic roles inferred by Arborist on the Mammal taxonomy. One-dimensional PCA projection (on the y-axis) of the node-embeddings $\mathbf{w}_v$ for all internal nodes v. Nodes colored by their assigned roles. Nodes assigned to the substance holonym role omitted due to rarity.

5.6 Inferred Taxonomic Roles

We now qualitatively explore the embeddings of each node $\mathbf{w}_v$ learned by Arborist. We are interested in whether these embeddings serve as a good proxy for the nodes' *taxonomic roles*, without being explicitly trained for taxonomic role classification. We exploit the availability of ground-truth edge-semantics available in the Mammal taxonomy to assign one of three taxonomic roles to each internal node: *hypernym* (is-a), *part-holonym* (is-part-of-whole) or *substance-holonym* (is-part-of-substance). We construct the assignment based on each nodes' most frequent incoming edge type. Statistics of the assignment are reported in Table 6. Due to the rarity of the *substance-holonym* role, we omit it from future consideration in this section.

To understand the variation in node-embeddings with their ground-truth taxonomic roles, we visualize in Fig. 4 the 1-dimensional PCA projections (on the y-axis) of $\mathbf{w}_v$ for all the internal nodes of the Mammal taxonomy. Nodes are colored based on their assigned taxonomic role, and a random sample of nodes are labelled by their associated text. While we do not observe any distinct clustering (even with two-dimensional PCA or t-SNE projections), we do observe a systematic variation in the taxonomic-role with the node-embedding. Fig. 4 reveals a continuous transition of taxonomic roles along the PCA dimension (the y-axis) from the *hypernym* role (at the bottom) to the *part-holonym* role (at the top).

Node Role	Number of nodes
Hypernym (IS-A)	1164
Part-Holonym (IS-PART-OF-WHOLE)	260
Substance-Holonym (IS-PART-OF-SUBSTANCE)	13

Table 6: Number of internal nodes in MAMMAL assigned to each role, based on their most frequent incoming edge-type.

Role-Classifier	ROC-AUC (%)	F1-Score (%)	Accuracy (%)
Majority Class	50.000	32.000	80.946
Random Forest	76.001	44.536	81.641

Table 7: Performance of node-role classification using the embeddings $\mathbf{w}_v$ learned by ARBORIST. Nodes restricted to internal nodes with hypernym or part-holonym roles, split into 50% train and test subsets. Thresholds for F1-score and accuracy tuned on the test subset.

This suggests that, while simple clustering may not be able to recover taxonomic roles from node-embeddings, more complex models may succeed. To quantify if any signal exists in the node-embedding that predicts its taxonomic role, we train a Random Forest classifier with 100 trees on the node-embeddings and report the taxonomic-role classification performance in Table 7. The classifier performs significantly better than simply predicting the majority class (*hypernym*). This confirms the existence of some predictive signal in the node-embedding for taxonomic-role classification. We delegate further study on the recovery of taxonomic-roles from node embeddings to future work.

6 CONCLUSION

We have proposed ARBORIST, an approach to automatically expand textual taxonomies with heterogeneous edge semantics that are unobserved. ARBORIST learns latent representations of the edge semantics, along with node embeddings capturing their taxonomic roles, to measure taxonomic relatedness between node-pairs. ARBORIST is trained by optimizing a large-margin ranking loss with a dynamic margin function, which is theoretically guaranteed to minimize an upper-bound on the shortest-path distance between the predicted parents and actual parents in the taxonomy. The shared linear maps representing the edge semantics and large-margin ranking loss improve robustness and generalization. Via extensive evaluation on a large-scale taxonomy at Pinterest and several public datasets, we have demonstrated that ARBORIST outperforms the state-of-the-art. The inferred taxonomic roles further induce a useful categorization of the taxonomy nodes that merits further exploration. In future work, we plan to develop an online update mechanism to train ARBORIST in a streaming fashion. We also plan to derive confidence scores for the predicted parents, providing an additional knob for human experts to filter out irrelevant predictions.

ACKNOWLEDGMENTS

The authors thank John Milinovich, Yunsong Guo, Fei Liu, Heath Vinicombe and all participants of the Pinterest ML Lunch for useful advice and feedback.

APPENDIX: IMPLEMENTATION DETAILS

We describe in this appendix several implementation details that we found were crucial for performance.

6.1 Dichotomous node feature-vectors

For each node v, we create two versions of its feature-vector, $\mathbf{e}_v^1$ to use when the node appears in eq. (2) as a child and $\mathbf{e}_v^2$ to use when it appears as a parent. We fix $\mathbf{e}_v^1$ but allow $\mathbf{e}_v^2$ to be optimized during training. This is analogous to the pivot and context vectors in word2vec [39].

6.2 Parameter initialization

We adopt the following initialization strategy for both ARBORIST and CRIM. Node feature-vectors are normalized to have unit L_2 norm before training (but are allowed to have unrestricted norms henceforth). The linear maps $P_1, \ldots, P_k$ are initialized to the identity matrix plus zero-mean Gaussian noise with standard-deviation 0.01 (as in [9]). For ARBORIST, we only construct M_v for internal nodes in the training taxonomy, since leaf nodes have no incoming edges to learn from. This significantly reduces the number of parameters (in a complete binary tree, leaf nodes account for over half the nodes in the tree).

6.3 Margin pre-computation

We pre-compute and store the margins $\gamma(u, v, v') = d(v, v')$ by computing the the undirected shortest-path distance between all pairs of nodes in the training taxonomy using breadth-first search in $O(V^2 + VE)$ time. We set the distance between disconnected nodes to the maximum shortest-path distance between any two nodes in the training taxonomy $\mathrm{d}_{\max}$.

6.4 Score and margin scaling

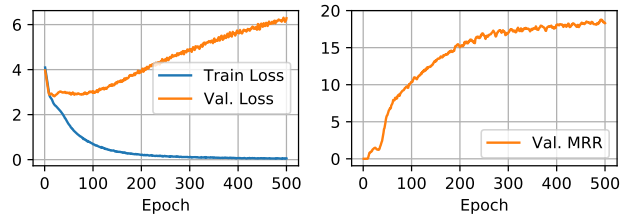
Scores $s(u, v)$ are unbounded, while margins $\gamma(u, v, v') = d(v, v')$ are typically small non-negative integers. Hence, we scale the score to lie in $[0, 1]$ by applying a sigmoid transformation. We also scale the margin to lie in $[0, 1]$ by normalizing it by $\mathrm{d}_{\max}$, the maximum distance between any two nodes. This causes the margin-violations (eq. 6) to become extremely small floating-point values and lead to numerical instabilities. To alleviate this issue, both the score and margin are further log-transformed.

6.5 Cross-validation

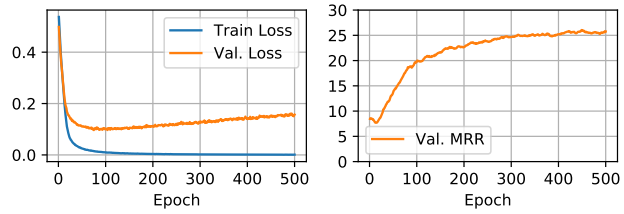
We found the validation loss to be a poor proxy for taxonomy expansion performance (exemplified by the loss curves in Fig. 5). Hence, we use the validation MRR for model selection instead.

6.6 Hyperparameters

Both ARBORIST and CRIM benefit from large batch-sizes (limited by available memory), low learning rates (limited by available time), and a large number of linear-maps (limited by the available memory). CRIM is sensitive to the number of negative samples, while ARBORIST is robust (due to the margin loss). Both ARBORIST and CRIM and are insensitive to dropout and L_2 regularization.



(a) CRIM on Mammal



(b) Arborist on Mammal

Figure 5: Train/validation loss curves (left) and validation MRR with epoch (right) of CRIM and Arborist on Mammal.

To ensure fair comparison, we use the same number of projections for both Arborist and CRIM in each dataset. Hyperparameter values are reported in Table 8.

6.7 Input Embeddings

300-dimensional FastText embeddings were constructed using the pre-trained `crawl-300d-2M-subword` model. 300-dimensional PinText embeddings were constructed in-house based on the training procedure described in [80]. Embeddings for out-of-vocabulary words were inferred using subword information. Embeddings for multi-word phrases were constructed by averaging the embeddings of the words in the phrase.

6.8 Hardware

Experiments were run on an Intel Xeon E5-2670 machine at 2.30GHz with 48 cores and 256GB of main memory. Implementations were in PyTorch 1.2.0 / Python 2. Training was performed only on CPUs.

Dataset	Pinterest	SemEval	Mammal
Arborist			
No. of train epochs	150	150	500
No. of projections	48	64	48
No. of neg. samples	5	10	5
Batch-size	4096	11000	4096
Learning-rate	0.0005	0.0005	0.0005
Weight-decay (l_2 penelty)	0.001	0.01	0.001
CRIM			
No. of train epochs	150	150	500
No. of projections	48	64	48
No. of neg. samples	5	1	5
Batch-size	10240	11000	4096
Learning-rate	0.005	0.0005	0.001
Weight-decay (l_2 penelty)	0.001	0.001	0.001

Table 8: Hyperparameter settings.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *WSDM*.
[2] Nesreen K Ahmed, Ryan Rossi, John Boaz Lee, Theodore L Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. 2018. Learning role-based graph embeddings. In *StarAI workshop at IJCAI*.
[3] 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 *ACL*.
[4] Tuan Luu Anh, Yi Tay, Siu Cheung Hui, and See Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *EMNLP*.
[5] Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Massih R Amini. 2013. On flat versus hierarchical classification in large-scale taxonomies. In *NIPS*.
[6] Rohit Babbar, Ioannis Partalas, Eric Gaussier, Massih-Reza Amini, and Cécile Amblard. 2016. Learning taxonomy adaptation in large-scale classification. *JMLR* (2016).
[7] Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *ACL*.
[8] Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *EACL*.
[9] Gabriel Bernier-Colborne and Caroline Barriere. 2018. CRIM at SemEval-2018 Task 9: A Hybrid Approach to Hypernym Discovery. In *SemEval*.
[10] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *TACL* (2017).
[11] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.
[12] Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 task 9: Hypernym discovery. In *SemEval*.
[13] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. 2018. Representation Tradeoffs for Hyperbolic Embeddings. In *ICML*.
[14] Ofer Dekel, Joseph Keshet, and Yoram Singer. 2004. Large margin hierarchical classification. In *ICML*.
[15] Tiansi Dong, Chrisitan Bauckhage, Hailong Jin, Juanzi Li, Olaf Cremers, Daniel Speicher, Armin B Cremers, and Jörg Zimmermann. 2018. Imposing Category Trees onto Word-Embeddings Using a Geometric Construction. (2018).
[16] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning structural node embeddings via diffusion wavelets. In *KDD*.
[17] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *WWW*.
[18] Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *EMNLP*.
[19] Yushu Feng, Huan Wang, Roland Hu, and Daniel T Yi. 2019. Triplet Distillation for Deep Face Recognition. In *ICML 2019*.
[20] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *ACL*.
[21] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*.
[22] Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *ACL*.
[23] Rafael S Gonçalves, Matthew Horridge, Rui Li, Yu Liu, Mark A Musen, Csongor I Nyulas, Evelyn Obamos, Dhananjay Shrouty, and David Temple. 2019. Use of OWL and Semantic Web Technologies at Pinterest. In *ISWC*. Springer.
[24] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
[25] Ruining He, Chunbin Lin, Jianguo Wang, and Julian McAuley. 2016. Sherlock: sparse hierarchical embeddings for visually-aware one-class collaborative filtering. In *AAAI*.
[26] Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *ACL*.
[27] 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. In *WSDM*.
[28] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. 2012. Supercharging recommender systems using taxonomies

for learning user purchase behavior. In *VLDB*.
[29] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[30] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations?. In *NAACL-HLT*.
[31] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. 2018. Smoothing the Geometry of Probabilistic Box Embeddings. (2018).
[32] Daryl Lim and Gert Lanckriet. 2014. Efficient learning of mahalanobis metrics for ranking. In *ICML*.
[33] David C Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related pins at Pinterest: The evolution of a real-world recommender system. In *WWW*.
[34] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. 2018. On interpretation of network embedding via taxonomy induction. In *KDD*.
[35] Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-End Reinforcement Learning for Automatic Taxonomy Induction. In *SDM*.
[36] Andrew McCallum, Ronald Rosenfeld, Tom M Mitchell, and Andrew Y Ng. 1998. Improving Text Classification by Shrinkage in a Hierarchy of Classes.. In *ICML*.
[37] Brian McFee and Gert R Lanckriet. 2010. Metric learning to rank. In *ICML*.
[38] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. 2011. Response prediction using collaborative filtering with hierarchies and side-information. In *KDD*.
[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.
[40] George A Miller. 1995. WordNet: a lexical database for English. *CACM* (1995).
[41] Farzaneh Mirzazadeh, Yuhong Guo, and Dale Schuurmans. 2014. Convex co-embedding. In *AAAI*.
[42] Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *ACL*.
[43] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *EMNLP*.
[44] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NIPS*.
[45] Maximilian Nickel and Douwe Kiela. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *ICML*.
[46] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *AAAI*.
[47] Alberto Paccanaro and Geoffrey E Hinton. 2002. Learning hierarchical structures with linear relational embedding. In *NIPS*.
[48] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
[49] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
[50] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *KDD*.
[51] Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*.
[52] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.
[53] Blake Shaw, Bert Huang, and Tony Jebara. 2011. Learning a distance metric from a network. In *NIPS*.
[54] Blake Shaw and Tony Jebara. 2009. Structure preserving embedding. In *ICML*.
[55] 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 *KDD*.
[56] Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *ACL*.
[57] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.
[58] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.
[59] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.
[60] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*.
[61] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.
[62] Dmitry Ustalov, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. 2017. Negative sampling improves hypernymy extraction based on projection learning. In *EACL*.
[63] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. In *ICLR*.
[64] Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. 2012. Learning hierarchical similarity metrics. In *CVPR*.
[65] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. In *ACL*.
[66] Luke Vilnis and Andrew McCallum. 2015. Word Representations via Gaussian Embedding. In *ICLR*.
[67] Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources. In *NAACL-HLT*.
[68] Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *EMNLP*.
[69] Jiayun Wang, Sanping Zhou, Jinjun Wang, and Qiqi Hou. 2018. Deep ranking model by large adaptive margin learning for person re-identification. *Pattern Recognition* (2018).
[70] Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *COLING*.
[71] Kilian Q Weinberger and Olivier Chapelle. 2009. Large margin taxonomy embedding for document categorization. In *NIPS*.
[72] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *JMLR* (2009).
[73] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *ICCV*.
[74] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. 2003. Distance metric learning with application to clustering with side-information. In *NIPS*.
[75] Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional hypernym generation by jointly learning clusters and projections. In *COLING*.
[76] Hui Yang and Jamie Callan. 2009. A metric-based framework for automatic taxonomy induction. In *ACL*.
[77] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning Term Embeddings for Hypernymy Identification.. In *IJCAI*.
[78] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *KDD*.
[79] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. 2014. Taxonomy discovery for personalized recommendation. In *WSDM*.
[80] Jinfeng Zhuang and Yu Liu. 2019. PinText: A Multitask Text Embedding System in Pinterest. In *KDD*.
[81] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *CIKM*.