Embedding Semantic Taxonomies

Alyssa Lees
Google Research
alyssalees@google.com

Chris Welty
Google Research
cawelty@gmail.com

Jacek Korycki
Google
korycki@gmail.com

Sara Mc Carthy
Google Research
sara.m.mccarthy@gmail.com

Shubin Zhao
Google Research
shubin@google.com

Abstract

A common step in developing an understanding of a vertical domain, e.g. shopping, dining, movies, medicine, etc., is curating a taxonomy of categories specific to the domain. These human created artifacts have been the subject of research in embeddings that attempt to encode aspects of the partial ordering property of taxonomies. We compare Box Embeddings, a natural containment-based representation of taxonomies, to partial-order embeddings and a baseline Bayes Net, in the context of representing the Medical Subject Headings (MeSH) taxonomy given a set of 300K PubMed articles with subject labels from MeSH. We deeply explore the experimental properties of training box embeddings, including preparation of the training data, sampling ratios and class balance, initialization strategies, and propose a fix to the original box objective. We then present first results in using these techniques for representing a bipartite learning problem (i.e. collaborative filtering) in the presence of taxonomic relations within each partition, inferring disease (anatomical) locations from their use as subject labels in journal articles. Our box model substantially outperforms all baselines for taxonomic reconstruction and bipartite relationship experiments. This performance improvement is observed both in overall accuracy and the weighted spread by true taxonomic depth.

1 Introduction

Recent work on hierarchical representational structures in machine learning promise to blend the value of human curated taxonomies with the power and flexibility of machine learning systems. A plethora of such taxonomies exist across many domains as seen in libraries, linguistic resources, medicine and popular culture, yet they rely on an assumption of discreteness – category membership is either true or false – and this assumption does not generally lend itself to modelling with continuous valued systems or spatial embeddings.

Taxonomic knowledge can be leveraged in the semantic space to extract hierarchical relations between words, and this simple observation has been the basis of many computational resources for linguistics, such as WordNet, EuroWordNet, FrameNet, etc. For example, the similarity of the words *lion* and *tiger* may be indistinguishable from the similarity of *lion* and *animal* in a naive text embedding space, as they may simply be the same distance apart, but in WordNet the difference and similarity are made clear through the taxonomy. In many cases, this explicit knowledge can be derived from existing taxonomies, but without some way of understanding this knowledge in the embedding space, it is just another similarity signal. Recently, order embeddings addressed this issue by orienting similarity in two axes and adding a constraint to similarity embeddings that ensured more general terms had to be closer to the origin (Liu et al., 2012).

More recently we've seen the introduction of box lattice embeddings, which treat categories in a taxonomy as n-dimensional boxes, with a constraint that boxes representing more general terms should *contain* boxes representing more specific terms (Vilnis et al., 2018).

Proceedings of the 28th International Conference on Computational Linguistics, pages 1279–1291
Barcelona, Spain (Online), December 8-13, 2020

On a different front, there has been steady progress in automating knowledge-base construction (KBC) from text, as a pure demonstration of NLP as well as a productive tool for end to end tasks like question answering. A key part of KBC efforts is learning relations from text, yet a key research question remains relatively unexplored: can we use taxonomic similarity in conjunction with KB relations? In the early days of NLP and knowledge representation (KR), it had always been assumed these two kinds of knowledge would work together, but there is little evidence in existing systems that they do. This work is motivated by that key research question.

Similar to prior work, we began by experimenting with various taxonomy embedding techniques to reconstruct the Medical Subject Headings (MeSH) taxonomy from published article tags. We customize a Box Embedding model for the task and compare the results to several baselines. Finally, we explore a familiar problem: learning a relation on a bipartite graph, in this case the relation between diseases and parts of the body using the co-occurrence of human-assigned MeSH labels on PubMed abstracts. The work tests the hypothesis that the MeSH taxonomy can provide a useful signal in learning this relation: if asthma (disease) and bronchi (anatomy) co-occur in data, and bronchi is a subcategory of lung in the taxonomy, then asthma and lung are related.

The contributions of this paper include the following:

1. A customized box model objective function that smooths the non-differentiable hinge property of the naive learning problem
2. Performance improvements for the taxonomic learning task via variations of negative sampling and weighting.
3. A methodology for incorporating sparse instance data evidence without exploding the size of the parameter space
4. Defining evaluation metrics specialized to the unique requirements of the learning of a taxonomy
5. Defining and testing novel experiments for learning taxonomies on bipartite graphs.

2 Related Work

Previous research in learning semantic taxonomies has included a few approaches. (Cimiano et al., 2005) proposed learning hierarchies from straight text using Formal Concept Analysis and (Maedche and Staab, 2001) learned ontologies with the semantic web. (Liu et al., 2012) extracted taxonomies from keywords. (Hoxha et al., 2016) learned taxonomy by clustering based on informed syntactic matching and semantic relations. (Velardi et al., 2013) automatically extracts taxonomies from web sites using concepts and hypernym relations.

Much early work with ontologies focused explicitly on expanding a given taxonomy or examining a specific area in more depth. (Snow et al., 2005) used a classifier for predicting new undiscovered edges in WordNet. (Kozareva, 2010) used an initial taxonomy to algorithmically crawl the web. (Shen et al., 2018) uses a seed taxonomy for guided hierarchical expansion. Other work considered noisy real word graphs and suggested algorithms to remove edges from a formal taxonomy (Velardi et al., 2013).

Similarly, there has been a corpus of work in embedding techniques to extract hierarchical structures in text. (Globerson et al., 2007) utilized co-occurrence for creating simple euclidean embeddings. (Zhang et al., 2018) employs hierarchical clustering in a recursive process to construct topic taxonomies. (Vendrov et al., 2016) represented text with partial order embeddings (POE). (Li et al., 2018) explored partial order embeddings with distributional co-occurrences for learning ontologies. (Vilnis et al., 2018) extended the functionality of POE to better reflect conditional probabilities of an ontology with Box embeddings. (Athiwaratkun and Wilson, 2018) explored Gaussian word embeddings. Earlier the authors showed preliminary results using Box Embeddings for constructing taxonomies in (Lees et al., 2019). In this work, we expand the performance analysis and add experimental results for the novel bipartite relations problem.

More recently, several works have explored hyperbolic embeddings, in which hierarchical structures are represented as continuous generalizations in non-Euclidean space. (Dhingra et al., 2018) (Nickel and Douwe, 2017) (Nickel and Douwe, 2018) (Ganea et al., 2018) (De Sa et al., 2017). (Le et al., 2017) uses hyperbolic embeddings for learning concept hierarchies from text. (Tifrea et al., 2019) introduced an application of GLOVE in hyperbolic space.

3 MeSH Dataset

MeSH, the NLM *Medical Subject Headings*, (U.S. National Library of Medicine, 2018a) is a taxonomy of subject headings for categorizing medical writing. Pubmed (U.S. National Library of Medicine, 2018b) is a very large collection over 30 million medical journal articles, each with metadata including human labeled subject categories from MeSH.

MeSH is organized into 16 top level categories, such as `A: Anatomy`, `B: Organisms`, `C: Diseases`, which themselves cannot be the subjects of articles. The taxonomy is on average 8 levels deep, with a generalization or broader-term relationship from child to parent nodes, e.g. `<Respiratory System, Anatomy>`, `<Larynx, Respiratory System>`, `<Asthma, Diseases>`. We adopt the notation in which the taxonomy is represented as a collection of $\langle child, parent\rangle$ *edges* in a tree-shaped taxonomy graph (noting that it is not strictly a tree).

The MeSH anatomy hierarchy mostly follows a Mereological generalization (sub-parts to parts), while diseases follow a slightly more causal generalization (`<Pneumonia, Bacterial Infection>`). This kind of semantic promiscuity is extremely common in taxonomies (Guarino and Welty, 2009), and causes an imprecision that begs for an approach with soft, continuous-valued constraints, as opposed to the traditional discrete reading of the taxonomic relationship (all members of the subcategory are members of the super-category). It was this observation that led us to taxonomy embeddings.

Articles listed in PubMed can have any number of subject headings, on average 8-10, and it is fully expected by the published methodology that assigning a particular subject heading to an article *also assigns the MeSH parents and ancestors* – they expect the transitive closure to hold.

4 Taxonomy Embedding Experiments

This work explores Box Lattice Embeddings (Vilnis et al., 2018) as a technique uniquely suited to the semantic taxonomy space. The efficacy of this approach is validated with Partial Order Embeddings and Naive Bayes are examined as baselines.

4.1 Partial Order Embeddings

We applied Partial Order Embeddings to our datasets, an established technique for modeling taxonomy data, as described in (Vendrov et al., 2016). The model assigns each entity u an embedding $f(u) \in \Re^n$ in such a way that the order in the space of entities, defined by edges $\langle u, v\rangle$, is maintained in the embedding space by requiring that $f(u) \geq f(v)$, which holds for all dimensions. This is accomplished by defining a continuous score function for embeddings that measures compliance with the order constraint:

$$E(x, y) = ||max(0, y - x)||^2$$

Note that max ranges over vector elements. This score should be zero, or close to it, for a pair forming an edge (from a set of positive examples P) and large positive for a pair that is not an edge (from a set of negative examples N). This requirement is enforced by minimizing the following max-margin loss function over embeddings f:

$$\sum_{(u,v)\in P} w(u, v)E(f(u), f(v)) + W \sum_{(u',v')\in N} max(0, \alpha - E(f(u'), f(v')))$$

It uses a margin $\alpha > 0$ to express the minimum desired value for a score of the negative example. Relative importance of positive examples may be controlled with edge weights w, if such values are available in the dataset (for example as conditional probabilities). Hyper-parameter W can be used to control relative importance of positive and negative parts of the loss.

4.2 Box Lattice Embeddings

Box lattice embeddings associate each category with 2 vectors in [0, 1], (x_m, x_M), the minimum and maximum value of the box at each dimension (Vilnis et al., 2018). For numerical reasons these are stored as a minimum, a positive offset plus an ϵ term to prevent boxes from becoming too small. This representation of boxes in Cartesian space can define a partial ordering by containment of boxes, and a lattice structure as:

$x \wedge y = \perp$ if x and y disjoint, else $\prod_i [max(x_{m,i}, y_{m,i}), min(x_{M,i}, y_{M,i})]$
$x \vee y = \prod_i [min(x_{m,i}, y_{m,i}), max(x_{M,i}, y_{M,i})]$

If $A = (x_m, x_M)$ then $p(A) = \prod_i^n (x_{M,i} - x_{m,i})$ and the objective function is

$$\Phi = \sum -log(p(a \vee b) - p(a) - p(b))$$

In other words, maximize the overlap of boxes with a positive edge. The original boxes paper showed results for reconstructing a taxonomy derived from WordNet using the transitive closure of edges, and the team made the code available in Github, which we reused and modified for our experiments.

4.3 Bayes Nets

As discussed in the next section, the data set we've chosen has an abundance of instance data from which conditional probabilities between categories can be calculated, making a Bayesian model an obvious choice to predict whether an instance belongs to a category. Given the training data, the model computes the conditional probability $p(C_x|C_y)$ for any category pair (C_x, C_y) that exists in the training data. Categories co-occur if they appear as labels on the same PubMed article. During training we choose a threshold τ where $p(C_x|C_y) > \tau \rightarrow edge \langle C_x, C_y \rangle$ such that the likelihood of known taxonomy edges in the training set is maximized.

5 Learning Taxonomies

Many taxonomies are *extensional*: the categories organize sets of entities in some problem domain, such as movies, stores, restaurants, books, songs, etc. There is a fairly clear instance/category distinction in these cases and the number of instances vastly outweighs the number of categories. Such taxonomies have edges from instances to categories, *inst-cat* edges (e.g. `<Star Wars, Science-Fiction Movie>`), and edges between categories, *cat-cat* edges (e.g. `<Cult Science-Fiction Movie, Science-Fiction Movie>`). Some taxonomies are *intensional*: they have no instances, at least none represented in data. WordNet synsets, for example, are arranged in a taxonomy, and there are very few instance-like synsets in WordNet, referring e.g. to specific people and organizations, but for the most part, synsets like "amount of matter" have no clear extension. Such taxonomies have only cat-cat edges.

Many previous experiments on learning taxonomy embeddings were conducted on intensional taxonomies, with no instances, and the objective (for training and evaluation) was simply the number of correct cat-cat edges learned above some confidence threshold.

Our datasets are extensional and contain millions of instances: PubMed 2018 has over 30 million.

5.1 Learning from Instances

We tried many approaches to utilizing the rich extension of MeSH in PubMed articles. The most obvious was to treat each inst-cat edge as a part of the graph, and ignore the semantic differences with cat-cat edges. However, in our embedding techniques, edges are training examples and the vertices (e.g. each category or instance) are the learnable parameters, so this results in an explosion of parameters almost to the point of having fewer examples than parameters. Further, instances as embeddings causes many spurious inst-inst edges to be inferred from the dense encodings, generating tremendous noise. One solution may be to use different forms of optimization, such as annealing or Brownian Motion. We save this for future work.

Another approach is to reuse the embedding techniques designed for cat-cat edges and *summarize* the inst-cat edges into the categories. For every pair of co-occurring inst-cat edges$\langle c, p_1 \rangle$ $\langle c, p_2 \rangle$, we emit a cat-cat edge $\langle p_1, p_2 \rangle$. This leads to repetition of edges in the training data that reflects the magnitude of the co-occurrence. We compared these two approaches:

Taxo: Use the edges from the taxonomy, with no consideration for the instances.

Summary: Include the co-occurrence of categories in instances as a weight on the taxonomy edges.

5.2 Transitive Closure

In addition to summarizing instances, we experimented with the use of deterministic reasoning on the training data, specifically the transitive closure of cat-cat and inst-cat edges. Given only *direct* cat-cat edges (e.g. if we have $\langle a, b \rangle$ and $\langle b, c \rangle$, and we do not have $\langle a, c \rangle$), we attempt to learn embeddings that approximate the taxonomy. For instance edges, we can similarly compute the transitive closure in the usual way (e.g. if we have $\langle i, a \rangle$ and $\langle a, b \rangle$ then we add $\langle i, b \rangle$), and then summarize from the inferred inst-cat edges as described above. This gives us two more variations on data sets:

Direct: do not compute the transitive closure of the category edges. For PubMed articles, only the human labelled subject headings are used. This implies most instances will not have edges to a category and its ancestors. Reconstructing a taxonomy from only direct edges is expected to be an extremely hard problem to solve with embedding methods.

Closure: compute the transitive closure of the category edges. PubMed articles will have edges to each human supplied subject heading and all its ancestors. The closure multiplies the positive training data and is expected to be a simpler problem.

inst	an instance (PubMed article)
cat	a category (MeSH subject heading)
inst-cat	edge from an instance to a category
cat-cat	taxonomy edge from a subcategory to a category
taxo	dataset consisting of cat-cat edges from the MeSH taxonomy
bip	dataset consisting of bipartite edges from MeSH Disease to Anatomy categories
summary	dataset of cat-cat edges weighted by # of instances shared by the two categories
direct	dataset consisting of cat-cat or inst-cat edges, as specified in MeSH and Pubmed
closure	dataset of direct edges augmented by the transitive closure of MeSH categories

Table 1: Terms used in this paper.

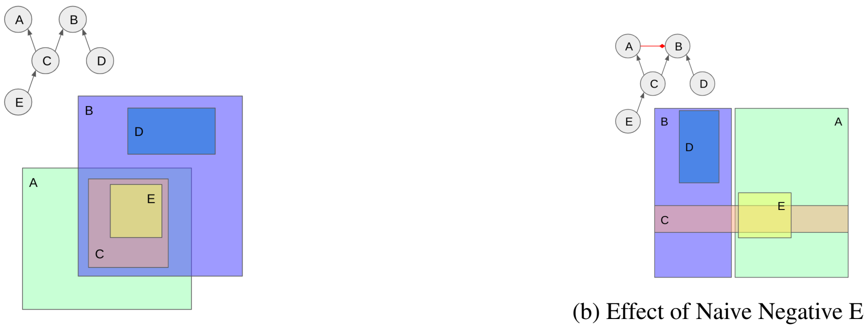
5.3 Baseline Taxonomy Reconstruction Experiments

With these parameters on our training data, we compared box embeddings, our bayes net, and order embeddings on their ability to reconstruct a 10% held out sample of the MeSH direct taxonomy edges, given the other 90% of edges, with F1 scores shown in Tab. 2. Instance summarization was on the first ten shards of PubMed 2018 (300k articles). While all edges in the test set are direct and held out, for every test edge $\langle a, b \rangle$, there must be edges in train containing a and b in order to form their embeddings.

Tag	Naive Box	Bayes	POE
taxo, direct	0.002	0.17	0.35
taxo, closure	0.00	0.77	0.83
negs, taxo, direct	0.002	–	0.63
negs, taxo, closure	0.00	0.78	0.89

Table 2: F1 scores of three taxonomy embedding approaches trained with taxonomy edges and tested on direct edges.

The experimental results in Table 2 demonstrate that Box Embeddings did not perform as expected. POE fared well, and seemed to respond well to the extra information provided by instance summarization and transitive closure. In most cases, the closure edges help the models classify the direct-only test edges.



(a) Idealized Box Rendering

(b) Effect of Naive Negative Edge

Figure 1: (a) An idealized box rendering of a simple taxonomy, and the actual learned box embedding. With no negative training data, all boxes simply overlap. (b) The effect of a naive negative edge – the two root categories are forced apart, and category C must split its loss between the two.

6 Customized Box Embeddings for Taxonomy Reconstruction

The promise of the representational power of boxes over vectors motivated us to dive deeper into the poor performance of boxes in these settings. Specifically, we modified the negative ratio, experimented with informed sampling, changed the objective function and utilized centered initialization to improve the box embedding model performance.

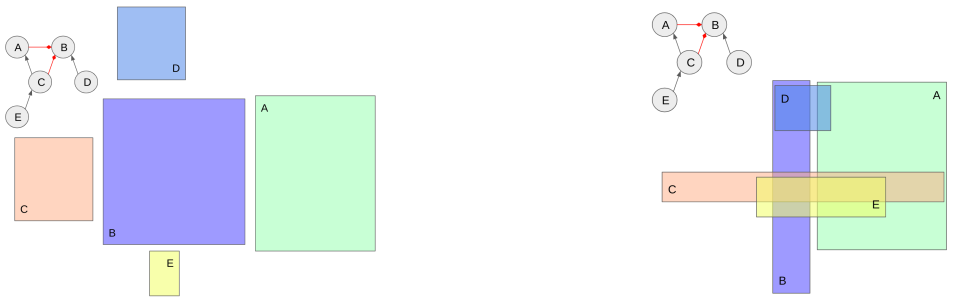
Negative Ratio: In the original box and order embedding papers, negatives were sampled uniformly from the complement of positive edges with a ratio of 1:1. The prior probability of a negative edge was greater than 0.99, so this sampling method allowed for a problem as shown in Fig. 1a. Given a simple toy taxonomy of five categories, Fig. 1a shows a version of the box layout we would expect, and Fig. 1b shows an actual learned set of box embeddings (in 2d) with no negatives. One problem becomes clear, the intuition of the box objective fails to account for the fact that it is easiest to satisfy the objective by making most of the boxes overlap.

Negative sampling rates are a common ML problem, and a commonly used approach is to increase the negative ratio to 10:1, which led to a problem illustrated in Fig. 1b. By increasing the number of negative edges in training, we increased the chances of a bad edge that creates competing constraints. In this case, since we have no edge $\langle A, B\rangle$ it could be naively sampled as a negative. Having that as a negative edge forces the two boxes apart, even though the edges $\langle C, A\rangle$ and $\langle C, B\rangle$ should cause them to overlap.

Informed sampling: Some improvements to sampling were proposed in (Athiwaratkun and Wilson, 2018), but these were more specific to order embeddings. Instead, we addressed the problems of naive sampling with *informed sampling* (see Appendix - in Section 10 for details). In a nutshell the algorithm uses traditional taxonomic reasoning as a way to identify edges that, while not necessarily positive, should not be negative. As in the example shown in Fig. 1b, one such case are edges between ancestors of the same category. Informed sampling improved matters but revealed another problem with training boxes, the *box crossing problem*, shown in Fig. 2a. In the example, we have altered the simple taxonomy a bit by making the edge $\langle C, B\rangle$ negative, and with A and C on opposite sides of B, the gradient will not allow them to cross and ultimately meet. As the neg:pos ratio increases, the chances of box crossing standing in the way increases dramatically. While in some sense this is no more than a gradient descent problem, getting stuck in a local neighborhood, the fact that boxes have volume, corners, and edges, leads to a solution space that is full of local minima. Solutions that smooth the space can help dramatically.

The improvements in performance from informed sampling can be seen for the Partial Order Embedding (POE) baselines in Table 2. Specifically, the rows marked with the *negs* designation outperform the same experiments without informed sampling.

Box Objective: The failure of the original box to generalize in the presence of more negatives forced us to discover and propose fixes to negative sampling, and the box crossing problem. However, ultimately we found the primary benefit came from modifying the hinge property of the loss function. When the box crossing problem arises, there is a non-differentiable hinge in the negative loss on the step where



(a) Box Crossing problem

(b) Fixed Box Objective

Figure 2: (a) The *box crossing* problem. Category C must cross the negative edge it has with B in order to get to A. In the original box model, box crossings cannot be overcome. (b) With the box objective fixed and a lower weight on negative, the crossing problem is solved by the thinned C piercing the thinned B to reach A.

the negative boxes first overlap. An approach to smoothing the loss has appeared very recently (Li et al., 2019). We used a far simpler fix developed before that result came out, which in combination with a different initialization strategy, resolved the box crossing problem.

In our modified loss function, positive and negative losses are defined with respect to the indicator function on a joint set of concept boxes a, b. For any given box embeddings we define the $meet(a, b)$ embedding of the boxes intersection volume and the *disjoint* indicator function if there is no intersection between the concept boxes. See Appendix 11 for details on meet function.

$$1_{[a,b]} = \begin{cases} 1 & \text{a} \wedge b = \perp \\ 0 & \text{otherwise} \end{cases}$$

Using the above notation we differentiate the instance losses for positive $f_{pos}(a, b)$, positive disjoint $f_{posd}(a, d)$ and negative loss $f_{neg}(a, b)$ scenarios. Definitions for these functions can be found in appendix 11. The objective modification ensures a smooth transition when boxes with a positive edge meet, and when boxes with a negative edge move apart. When positive boxes are disjoint and have a label indicating an overlap, there is loss proportional to their distance. This addition is to ensure that boxes that are far apart are encouraged to move closer together in physical space. When the boxes meet the loss is inversely proportional to the amount of overlap. At the point where they meet, these two kinds of loss must also meet, otherwise an unbounded gradient occurs.

$$L(\hat{a}_j, \hat{b}_j, l_j; \epsilon) = \begin{cases} \max(f_{pos}(a, b), -\log(\epsilon)) & \text{if } 1_{[a,b]} = 1 \text{ and } l_j > 0 \\ f_{posd}(a, b) - \log(\epsilon) & \text{if } 1_{[a,b]} = 0 \text{ and } l_j > 0 \\ \max(f_{neg}(a, b), -\log(\epsilon)) & \text{if } 1_{[a,b]} = 1 \text{ and } l_j = 0 \\ 0 & \text{otherwise} \end{cases}$$

For negative edges, when two overlapping boxes come apart, the negative loss indicated by $f_{neg}(a, b)$ is zero. The original box model clipped this value at ϵ in order to avoid a zero value, and added simple smoothing. This was not working in practice, as we saw sharp gradient spikes during optimization that had the effect of making negative boxes bounce away from each other. To solve this, we merely reversed the approach for positive loss, splitting the loss into overlapping and disjoint loss, and flipping the sense.

With these fixes in place, the new box models were able to overcome the box crossing problem as shown in Fig 2b.

Center init: In addition, while the solution proposed in (Li et al., 2019) uses a soft box boundary that effectively causes all boxes to overlap, we found that *center intitialization*, i.e. initializing all boxes to the center of the space, provided the same advantages – all boxes overlap at the beginning – while preserving the lattice properties of boxes. The original box model used random initialization. Center initialization provides, from analysis, a smoother gradient overall, leading to faster learning times.

6.1 Experimental Results

Combining a 10:1 negative ratio, informed sampling, a new objective, and center initialization, box models are able to model the taxonomy, as shown in Tab. 3. They are able to capitalize on the summary and taxonomy closure signals, and show better performance on this problem than the other methods.

Tag	Customized Box	Bayes	POE
taxo, direct	0.42	0.17	0.35
taxo, closure	0.89	0.78	0.83
summary, direct	0.97	0.82	0.89
summary, closure	**0.98**	0.80	0.91

Table 3: F1 scores of three taxonomy embedding approaches with improved box model.

7 Taxonomy Depth

The hierarchical nature of taxonomy datasets yields a further evaluation problem. When using box embeddings, it is expected that the volume of root nodes will substantially exceed those of the leaf nodes. As such, a naive evaluation of performance can yield decent results with a model that only learns the root nodes and/or nodes with edges close to the root. Given the structure of box embeddings where the marginal probability of a node is equivalent to the box volume, such an evaluation problem may be more pronounced. As such, we have examined the notion of *taxonomy depth* as used by (Yang and Callan, 2009).

Table 4 contains the F1 scores of the different models on summary + closure data at different levels of the taxonomy. The result demonstrates that decent learning metric scores were achieved for models that only learned superficial hierarchical structure. In other words, the Bayes and Order Embedding models' performance at higher levels of the taxonomy dominated their high overall F1, but for lower levels of the taxonomy, the performance was inconsistent. The Box Embedding model demonstrates stable performance at all depths, indicating the model learned the underlying taxonomy structure throughout. Consistent performance across depths is a crucial factor in the bipartite problem discussed in the next section.

Depth	Customized Box	Bayes	POE
1	1.00	0.81	1.00
2	**0.96**	0.62	0.89
3	**0.94**	0.58	0.87
4	**0.93**	0.57	0.86
5	**0.94**	0.58	0.86
6	**0.95**	0.57	0.88
7	**0.95**	0.50	0.85
8	**0.97**	0.49	0.87

Table 4: F1 scores of three taxonomy embedding approaches by taxonomy depth.

Tag	Customized Box	Bayes	POE
bip, sum10, direct	0.5	0.26	0.55
bip, sum10, closure	0.66	0.26	0.56
bip, sum100, direct	**0.7**	–	0.62
bip, sum100, closure	0.69	–	0.63

Table 5: F1 scores of three taxonomy embedding approaches on the bipartite (bip) relation problem using summaries from 10 and 100 shards of PubMed.

8 Bipartite Relations and Taxonomies

Given the ability to reconstruct a single taxonomy, we expand the problem definition to learn a bipartite relation between two distinct taxonomy branches of MeSH.

For simplicity, we focus on *A: Anatomy* and *C: Disease*, with a total of 6610 categories, with instance summarization from the first ten shards of the PubMed 2018 database (300k articles). The bipartite edges are formed through the summarization process, where the probability of each Disease to Anatomy edge $\langle d, a \rangle$ is the conditional probability $P(a|d)$ that a PubMed article is labelled with category a, given that it is labeled with d. Note that for the closure dataset, an article i is considered to be labeled with a category a_p if the following direct edges exist: $\langle i, a_c \rangle$, $\langle a_c, a_p \rangle$. As in previous experiments, we measure F1 on a held-out set of 10% of edges.

In these experiments, we attempt to jointly learn the in-taxonomy and cross-taxonomy relations between diseases and their anatomical locations. The intention is for the two taxonomies to overlay each other to represent the location relation.

For example, the *A:Anatomy* branch of the taxonomy contains the edge $\langle Heart, HeartValve \rangle$ and the *C:Disease* branch contains $\langle HeartValveDiseases, HeartDiseases \rangle$.

The underlying expectation is that co-occurrences of MeSH labels on PubMed articles will demonstrate that when an article mentions *Heart Valve Diseases* it will also mentions *Heart Valve*. This will establish a bipartite edge between the two *A:Anatomy* and *C:Disease* taxonomy branches. The hypothesis is that these direct co-occurrence edges will assist in extracting the more general edges such as $\langle HeartValveDisease, Heart \rangle$, $\langle HeartDisease, Heart \rangle$.

An initial analysis of the initial data revealed that many bipartite edges yielded very small marginal proportions. As such we conducted, a second set of experiment summarizing the edges using the first 100 shards of PubMed 2018 or roughly 10% of the total and roughly 30M articles.

Table 5 includes F1 scores for the bipartite (bip) learning problem. Sum10 refers to the original smaller PubMed instance summarization dataset and sum100 is the larger aggregated over 30 million articles. Our Naive Bayes system was too slow to finish sum100 in time for this submission.

The results were not as expected. The closure data, in which the transitive closure (ie the taxonomy) had been applied to the bipartite edges in the training data, showed no significant improvement over the direct (no closure) data. Increasing the size of the summarization did help improve overall representation, and box embeddings perform better than the other two methods.

An analysis of the learned embeddings discovered an exponential distribution of bipartite edge scores. With all three techniques, a threshold-setting step was applied specifying the minimum conditional probability on an edge in the embedding space for consideration as a discrete edge. However, it is acknowledged that for these bipartite relations, the embeddings are approximating the true conditional probabilities, which may or may not be a valid approximations for evaluating a *true* discrete edge. The exponential distribution of conditional probabilities indicates that the vast majority of co-occurrences are quite small. A better measure of the performance of these embeddings on the bipartite relation may be KL-divergence. We leave this investigation to future work.

9 Conclusions

We presented the practical problem of extracting implicit underlying taxonomies from instance data in the PubMed MeSH dataset. The work also explored the more complex task of extracting bipartite relations within multiple taxonomies.

An in-depth exploration of re-constructing a single taxonomy with Box Embeddings yielded experiments in negative sampling, solutions to the box crossing problem and a customized objective function. The final model demonstrated impressive F-measure scores for reconstructing the MeSH Disease taxonomy. Unlike other baselines, the model scores were consistent for all depths.

Finally, we defined a problem of extracting bipartite relations between multiple taxonomies within the PubMed article subject headings. Our customized box embedding models outperformed other baselines for both problems and we introduced a methodology conducive to the taxonomy extraction space.

References

Ben Athiwaratkun and Andrew Gordon Wilson. 2018. On modeling hierarchical data via probabilistic order embeddings. In *International Conference on Learning Representations*.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of artificial intelligence research*.

Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. 2017. Representation tradeoffs for hyperbolic embeddings. *NeurIps*.

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

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. *ICML*.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean embedding of co-occurrence data. *J. Mach. Learn. Res.*, 8:2265–2295, December.

Nicola Guarino and Christopher Welty, 2009. *An Overview of OntoClean*, pages 201–220. 05.

Julia Hoxha, Guoqian Jiang, and Chunhua Weng. 2016. Automated learning of domain taxonomies from text using background knowledge. *Journal of Biomedical Informatics*, 63:295 – 306.

Zornitsa Kozareva, Zornitsa Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. in proceedings of the 2010 conference on empirical methods in natural language processing. *ACL*.

Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2017. Inferring concept hierarchies from text corpora via hyperbolic embeddings. *ACL*.

A. Lees, J. Korycki, Chris Welty, and Shubin Zhao. 2019. Taxonomy embeddings on pubmed article subject headings. In *SEPDA@ISWC*.

Xiang Li, Luke Vilnis, and Andrew McCallum. 2018. Improved representation learning for predicting commonsense ontologies. *International Conference on Machine Learning Workshop on Deep Structured Prediction*.

Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. 2019. Smoothing the geometry of probabilistic box embeddings. In *International Conference on Learning Representations*.

Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. 2012. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1433–1441, New York, NY, USA. ACM.

Alexander Maedche and Steffen Staab. 2001. Ontology learning for the semantic web. *IEEE Intelligent systems*.

Maximillian Nickel and Kiela Douwe. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*.

Maximillian Nickel and Kiela Douwe. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *ICML*.

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*.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. *In Advances in neural information processing systems*.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. *ICLR*.

U.S. National Library of Medicine. 2018a. Mesh medical subject headings.

U.S. National Library of Medicine. 2018b. Pubmed medical abstracts.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *ICLR*.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *ACL*.

Grace Hui Yang and James P. Callan. 2009. A metric-based framework for automatic taxonomy induction. In *ACL/IJCNLP*.

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. KDD '18, page 2701–2709, New York, NY, USA. Association for Computing Machinery.

10 Appendix: Informed Negative Sampling

If negatives are generated on-the-fly (as they were in the original box embeddings implementation), negatives from the transitive closure can leak into training after a test/train split. We must generate negatives in batch and split into test/train.

Self edges should never be negative.

The inverse edges should never be negative, e.g. if $\langle c, p\rangle$ is a positive edge, $\langle p, c\rangle$ should not be negative. This is an interesting divergence between the box objective and the set semantics of categories: in a set semantics, the pair and its inverse can only be true if $c = p$, thus negating the inverse is equivalent to asserting strict subset. The box objective, however, pushes negative edges *apart*, and a sub-category must clearly overlap with its parent.

Edges between ancestors of the same category should never be negative, e.g. if $\langle c, p_1\rangle, \langle c, p_2\rangle$ are positive, then neither $\langle p_1, p_2\rangle$ nor its inverse should be negative since, in a strict reading of taxonomy, their boxes should overlap. In Fig. 1a, the idealized box embedding for a simple taxonomy is shown on the left; category C should be contained in both A and B, therefore the latter should overlap. Without negatives, the simplest zero-loss solution is to make all boxes overlap. In Fig. 1a the effect of a poorly chosen negative edge between A and B is shown, it forces the two boxes apart and C must deal with the loss. Again, in a set semantics, one may want to negate the subcategory edge between two overlapping categories, which would simply be interpreted as a constraint against one being a subset of the other. With boxes, doing so would again conflict with the objective to make them overlap.

We introduce *informed sampling* (Alg. 1) as a way of avoiding these problems. The algorithm incrementally builds $\mathcal{NN}$, the set of non-negative edges, by processing the constraints discussed above, and subtracts that from the edge set formed by the cross product of all categories in the taxonomy, $\mathcal{C}$.

Algorithm 1: Informed sampling of taxonomy negatives

```
Input: Set of positive edges $\mathcal{P} = \{\langle c, p\rangle, ...\}, c, p \in \mathcal{C}$
Output: Set of negative edges to sample from
$\mathcal{T} \leftarrow Closure(\mathcal{P})$
$\mathcal{NN} \leftarrow \mathcal{T}$
for $\langle c, p\rangle \in \mathcal{T}$ do
    $\mathcal{NN} \leftarrow \mathcal{NN} \cup \{\langle c, c\rangle, \langle p, p\rangle, \langle p, c\rangle\}$
    $map(c) \leftarrow map(c) \cup \{p\}$
end
for $c \in map$ do
    for $p_1 \in map(c)$ do
        for $p_2 \in map(c)$ do
            $\mathcal{NN} \leftarrow \mathcal{NN} \cup \{\langle p_1, p_2\rangle\}$
        end
    end
end
return $\mathcal{C} \times \mathcal{C} - \mathcal{NN}$
```

11 Appendix: Customized Box Embedding Objective

The customized Box Embedding Objective function is defined with respect to a joint set of concept boxes $\{a, b\}$ with corresponding box embeddings represented as a pair of vectors in $[0, 1]^n$ where n is number of dimensions, $(x_m, x_M), (y_m, y_M)$. A specific sampled instance is represented by $(\hat{a}_j, \hat{b}_j, l_j)$ where a label $l_j \in [0, 1]$ and represents proportional concept box overlap.

In our modified loss function, positive and negative loses are defined with respect to the indicator function. For any given box embeddings we define the *meet* embedding of the box intersection and the *disjoint* indicator function if there is no intersection $x \wedge y = \perp$.

$$meet(a,b) = (\max(x_{m,i}, y_{m,i}), \min(x_{M,i}, y_{M,i})) \forall i \in n$$

$$1_{[a,b]} = \begin{cases} 1 & \text{any } meet(a,b)_{M,i} <= meet(a,b)_{m,i} \\ 0 & otherwise \end{cases}$$

Using the above notation we differentiate the different instance loses for positive, positive disjoint and negative loss scenarios :

$$\begin{aligned} f_{pos}(a,b) = -\sum_i (\log(meet(a,b)_{M,i} - meet(a,b)_{m,i}) \\ - \log(x_{M,i} - x_{m,i})) \end{aligned}$$

$$\begin{aligned} f_{posd}(a,b) = -\sum_i \log(1 - (\max(meet(a,b)_{M,i}, \\ meet(a,b)_{m,i}) \\ - \min(meet(a,b)_{M,i}, meet(a,b)_{m,i}))) \end{aligned}$$

$$\begin{aligned} f_{neg}(a,b) = -\sum_i \log(1 - ((meet(a,b)_{M,i} - meet(a,b)_{m,i}) \\ - \log(x_{M,i} - x_{m,i}))) \end{aligned}$$

The full likelihood is defined with respect to the positive, postive disjoint and negative functions.

$$L(\hat{a}_j, \hat{b}_j, l_j; \epsilon) =$$

$$\begin{cases} \max(f_{pos}(a,b), -\log(\epsilon)) & \text{if } 1_{[a,b]} = 1 \text{ and } l_j > 0 \\ f_{posd}(a,b) - \log(\epsilon) & \text{if } 1_{[a,b]} = 0 \text{ and } l_j > 0 \\ \max(f_{neg}(a,b), -\log(\epsilon)) & \text{if } 1_{[a,b]} = 1 \text{ and } l_j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$