Inducing Hypernym Relationships Based On Order Theory

Sarthak Dash, Md Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya and Nicolas Rodolfo Fauceglia
IBM Research AI, Yorktown Heights, NY, USA
{sdash, mchowdh, gliozzo}@us.ibm.com
{nandana.m, nicolas.fauceglia}@ibm.com

Abstract
This paper introduces Strict Partial Order Networks (SPON), a novel neural network architecture designed to enforce asymmetry and transitive properties as soft constraints. We apply it to induce hypernymy relations by training with is-a pairs. We also present an augmented variant of SPON that can generalize type information learned for in-vocabulary terms to previously unseen ones. An extensive evaluation over eleven benchmarks across different tasks shows that SPON consistently either outperforms or attains the state of the art on all but one of these benchmarks.

1 Introduction
The ability to generalize the meaning of domain-specific terms is essential for many NLP applications. However, building taxonomies by hand for a new domain is time-consuming. This drives the requirement to develop automatic systems that are able to identify hypernymy relationships (i.e. is-a relations) from text.

Hypernymy relation is reflexive and transitive but not symmetric (Miller et al. 1990; Hearst 1992). For example, if Wittgenstein $\prec$ philosopher and philosopher $\prec$ person, where $\prec$ means is-a, it follows that Wittgenstein $\prec$ person (transitivity). In addition, it also follows that both philosopher $\not\prec$ Wittgenstein and person $\not\prec$ philosopher (asymmetry). Absence of self-loops within taxonomies (e.g. WordNet (Miller et al. 1990)) emphasizes that reflexivity (e.g. person $\prec$ person) does not add any new information.

In order theory, a partial order is a binary relation that is transitive, reflexive and anti-symmetric. A strict partial order is a binary relation that is transitive, irreflexive and asymmetric. Strict partial orders correspond more directly to directed acyclic graphs (DAGs). In fact, hypernymy relation hierarchy in WordNet is a DAG (Suchanek, Kasneci, and Weikum 2008). Therefore, we hypothesize that the Hypernymy relations within a taxonomy can be better represented via strict partial order relations.

In this paper we introduce Strict Partial Order Networks (SPON), a neural network architecture comprising of non-negative activations and residual connections designed to enforce strict partial order as a soft constraint. We present an implementation of SPON designed to learn is-a relations. The input of SPON is a list of is-a pairs, provided either by applying Hearst-like patterns over a text corpus or a list of manually validated pairs.

In order to identify hypernyms for out-of-vocabulary (OOV) terms, i.e. terms that are not seen by SPON during the training phase, we present an augmented variant of SPON that can generalize type information learned for the in-vocabulary terms to previously unseen ones. The augmented model does so by using normalized distributional similarity values as weights within a probabilistic model, the details of which are described in Section 5.

The main contributions of this paper are the following:

• We introduce the idea of Strict Partial Order Network (SPON), highlighting differences and similarities with previous approaches aimed at the same task.
• A theoretical analysis shows SPON enforces asymmetry and transitivity requirement as soft constraints.
• An augmented variant of SPON to predict hypernyms for OOV is proposed.
• Compared to previous approaches, we demonstrate that our system achieves and/or improves the state of the art (SOTA) consistently across a large variety of hypernymy tasks and datasets (multi-lingual and domain-specific), including supervised and unsupervised settings.

The rest of the paper is structured as follows. Section 2 describes related work. SPON is introduced in Section 3, and theoretical analysis is provided in Section 4. In Section 5 we show how SPON can be augmented for OOV terms in the test dataset. Section 6 and 7 describe the evaluation setup and results. Section 8 concludes the paper and highlights perspectives for future work.

2 Related Work
Since the pioneering work of Hearst (1992), lexico-syntactic pattern-based approaches (e.g., “NP$_y$ is a NP$_x$”) remains influential in subsequent academic and commercial applications. Some work tried to learn such patterns automatically (Snow, Jurafsky, and Ng 2004; Shwartz, Goldberg, and Dagan 2016) instead of using a predefined list of patterns.

Among other notable work, Kruszewski, Paperno, and Baroni (2015) proposed to map concepts into a boolean
lattice. Lin and Pantel (2002) approached the problem by clustering entities. Dalvi, Cohen, and Callan (2012) proposed to combine clustering with Hearst-like patterns. There also exist approaches (Weeds, Weir, and McCarthy 2004; Roller and Erk 2016; Shwartz, Santus, and Schlechtweg 2017) inspired by the Distributional Inclusion Hypothesis (D IH) (Zhitomirsky-Geffet and Dagan 2005).

Fu et al. (2014) argued that hypernym-hyponym pairs preserve linguistic regularities such as \( v(\text{shrimp}) \approx v(\text{prawn}) \approx v(\text{fish}) \approx v(\text{goldfish}) \), where \( v(w) \) is the embedding of the word \( w \). In other words, they claimed that a hyperonym word can be projected to its hypernym word learning a transition matrix \( \Phi \). Tan, Gupta, and van Genabith (2015) proposed a deep neural network based approach to learn \( is-a \) vectors that can replace \( \Phi \).

Recently, Roller, Kiela, and Nickel (2018) showed that exploitation of matrix factorization (MF) on a Hearst-like pattern-based system’s output vastly improved their results (for different hypernymy tasks; in multiple datasets) with comparison to that of both distributional and non-MF pattern-based approaches.

Another thread of related work involves the use of graph embedding techniques for representing a hierarchical structure. Order-embeddings (Vendrov et al. 2016) encode text and images with embeddings, preserving a partial order (i.e. \( x \preceq y \), where \( x \) is a specific concept and \( y \) is a more general concept) over individual embedding dimensions using the Reversed Product Order on \( \mathbb{R}^{d} \). In contrast, our proposed neural network based model encodes a strict partial order through a composition of non-linearities and residual connections. This allows our model to be as expressive as possible, all the while maintaining strict partial order.

Li, Vilnis, and McCallum (2017) extended the work of Vendrov et al. (2016) by augmenting distributional co-occurrences with order embeddings. In addition, hyperbolic embeddings model tree structures using non-euclidean geometries, and can be viewed as a continuous generalization of the same (Nickel and Kiela 2017). Other recent works have induced hierarchies using box-lattice structures (Vilnis et al. 2018) and Gaussian Word Embeddings (Athiwaratkun and Wilson 2018).

Regarding the recent SOTA, for unsupervised setting where manually annotated (i.e. gold standard) training data is not provided, Le et al. (2019) proposed a new method combining hyperbolic embeddings and Hearst-like patterns, and obtained significantly better results on several benchmark datasets.

For supervised setting, during the SemEval-2018 hypernymy shared task (Camacho-Collados et al. 2018b), the CRIM system (Bernier-Colborne and Barriere 2018) obtained best results on English datasets (General English, Medical and Music). This system combines supervised projection learning with a Hearst-like pattern-based system’s output. In the same shared task, for Italian, the best system, \( 300-\text{sparsans} \), was a logistic regression model based on sparse coding and a formal concept hierarchy obtained from word embeddings (Berend, Makrai, and Földiák 2018); whereas for Spanish, the best system, \( \text{NLPP}_{\text{HZ}} \) was based on the nearest neighbors algorithm (Qiu et al. 2018).

In Sections 6 and 7 we compare our approach with all of the above mentioned recent SOTA in both unsupervised and supervised settings, respectively.

### 3 Strict Partial Order Networks

The goal of SPON is to estimate the probability for a distinct pair of elements \( x, y \in \mathcal{E} \) to be related by a strict partial order \( x \preceq y \). A specific instance of this problem is the hypernym detection problem, where \( \mathcal{E} \) is a vocabulary of terms and \( \preceq \) is the \( is-a \) relation. In this section, we present a SPON implementation, while a theoretical analysis of how the proposed architecture satisfies transitive and asymmetric properties is described in the next section.

An implementation of a SPON is illustrated in Figure 1. Each term \( x \in \mathcal{E} \) is represented via a vector \( \vec{x} \in \mathbb{R}^{d} \). In the first step, we perform an element-wise multiplication with a weight vector \( w_{1} \) and then add to a bias vector \( b_{1} \). The next step consists of a standard ReLU layer, that applies the transformation \( \text{ReLU}(v) = \max(0, v) \). Let us denote these transformations by a smooth function \( g \),

\[
g(\vec{x}) = \text{ReLU}(w_{1} \odot \vec{x} + b_{1})
\]

where \( \odot \) denote element-wise multiplication.

The final step, as depicted in Figure 1, consists of a residual connection, i.e.

\[
f(\vec{x}) = \vec{x} + g(\vec{x})
\]

![Simple SPON architecture](image)

**Figure 1:** Simple SPON architecture.

We encode the loss layer to capture the distance-to-satisfaction \( \psi \) for a given candidate hyponym-hypernym pair \( (x, y) \), defined as follows:

\[
\psi(x, y) = \sum_{i=1}^{d} \max(0, \epsilon + f(\vec{x})_{i} - y_{i})
\]

where the sum is taken over all the components of the participating dimensions, and \( \epsilon \) is a scalar hyper-parameter.

The network is trained by feeding positive and negative examples derived from a training set \( T \) containing \( is-a \) relations and their corresponding scores. Each positive training
instance consists of a pair \((x, \mathcal{H}_x)\), where \(\mathcal{H}_x\) is the set of candidate hypernyms of \(x\) in the training data. Negative instances for a given term \(x\), denoted by \(\mathcal{H}'_x\), are generated by selecting terms uniformly at random from \(\mathcal{E}\). More formally, for a given candidate hyponym term \(x\), let
\[
\mathcal{H}_x = \{ e \in \mathcal{E} \mid (x, e, s) \in \mathcal{T} \} \tag{4}
\]
denote all the candidate hypernym terms of \(x\), and let
\[
\mathcal{H}'_x = \{ e \in \mathcal{E} | (x, e, s) \notin \mathcal{T} \} \tag{5}
\]
denote negative hypernym samples for \(x\). Negative hypernym terms are sampled at random from \(\mathcal{E}\), and as many negative samples are generated that satisfy \(|\mathcal{H}_x| + |\mathcal{H}'_x| = k\), a constant (hyper-parameter for the model).

The probability of \((x, y)\) being a true hyponym-hypernym pair is then calculated using an approach analogous to Boltzmann distribution as follows,
\[
p(x, y) = \frac{e^{-\psi(x,y)}}{\sum_{z \in \mathcal{H}_x \cup \mathcal{H}'_x} e^{-\psi(x,z)}} \tag{6}
\]
Equation 6 is used for training, while during scoring, the probability that a pair \((x, y)\) exhibits hypernymy relationship is given by,
\[
p(x, y) = \frac{e^{-\psi(x,y)}}{\sum_{z \in \mathcal{H}} e^{-\psi(x,z)}} \tag{7}
\]
whereas, the most likely hypernym term \(y^*\) for a given hyponym term \(x\) is given by,
\[
y^* = \arg \max_{y \in \mathcal{H}} \frac{e^{-\psi(x,y)}}{\sum_{z \in \mathcal{H}} e^{-\psi(x,z)}} \tag{8}
\]
Here, \(\mathcal{H}\) denotes the list of all hypernym terms observed in the training set \(\mathcal{T}\).

Finally, we define the loss function \(\mathcal{J}\) using a weighted negative log-likelihood criterion (w-NLL) defined as follows,
\[
\mathcal{J} = - \sum_{(x,y,z) \in \mathcal{T}} s \log p(x,y) \tag{9}
\]
where \(s\) represents the relative importance of the loss associated with pair \((x,y)\) \(\in \mathcal{T}\).

4 Theoretical Analysis

Hypernymy relations within a taxonomy satisfy two properties: asymmetry and transitivity. The asymmetry property states that given two distinct terms \(x, y \in \mathcal{E}\), if \(x \prec y\) then \(y \not\prec x\). The transitive property states that given three distinct terms \(x, y, z \in \mathcal{E}\), if \(x \prec y\) and \(y \prec z\) then \(x \prec z\).

In this section we analytically demonstrate that the neural network architecture depicted in Fig. 1, whose forward pass expressions are given by equations 1 and 2, satisfy asymmetry and transitive properties.

As described by equation 3, our proposed model assigns a zero loss for a given hyponym-hypernym pair \((x, y)\) if the learned model satisfies \(f(\mathcal{e}) < \mathcal{y}\) element-wise. This formulation of the loss layer puts forth the following constraint that defines our model,
\[
x \prec y \iff f(\mathcal{e})_i < \mathcal{y}_i, \forall i \tag{10}
\]
In other words, the relation \(\prec\) is satisfied if and only if \(f(\mathcal{e}) < \mathcal{y}\), component-wise. In the rest of this section, we show that under the assumption of [10], our proposed model for hypernymy relation satisfies asymmetry and transitivity.

Theorem 4.1. Expression 10 satisfies asymmetry.

Proof. Let \(x \prec y\). Then, it follows expression 10 that \(f(\mathcal{e}) < \mathcal{y}\) component-wise. We need to show that \(y \not\prec x\). Using the definition of expression 10, it is enough to show \(f(\mathcal{y}) \geq \mathcal{e}\) component-wise.

Now, using equation 2, we have \(f(\mathcal{y}) = \mathcal{y} + g(\mathcal{y})\). From the definition of function \(g\), it is clear that \(g(\mathcal{e}) \geq 0\) component-wise. Thus, applying this inequality to the previous expression, we have \(f(\mathcal{y}) \geq \mathcal{e}\) component-wise. On similar lines, we can also show that \(f(\mathcal{e}) \geq \mathcal{e}\) component-wise.

We now have \(f(\mathcal{y}) \geq \mathcal{e} > f(\mathcal{e}) \geq \mathcal{e}\) component wise. The middle inequality holds, since we assume \(x \prec y\); in other words, \(f(\mathcal{e}) < \mathcal{y}\) holds component wise. Thus expression 10 satisfies asymmetry.

\[\square\]

Theorem 4.2. Expression 10 satisfies transitivity.

Proof. Let \(x \prec y\) and \(y \prec z\). Then, it follows from expression 10 that \(f(\mathcal{e}) < \mathcal{y}\) and \(f(\mathcal{y}) < \mathcal{z}\), component wise. We need to show that \(x \prec z\) or, alternatively, that \(f(\mathcal{e}) < \mathcal{z}\).

Generalizing equation 11, we have, \(\forall e \in \mathcal{E}, f(\mathcal{e}) \geq \mathcal{e}\) component-wise. Using this observation, we have \(f(\mathcal{e}) < \mathcal{y} \leq f(\mathcal{y}) < \mathcal{z}\) component-wise. Note that the middle inequality holds from the aforementioned observation. This proves that Expression 10 satisfies transitivity.

\[\square\]

5 Generalizing SPON to OOV

The proposed SPON model is able to learn embedding for terms appearing in the training data (extracted either using Hearst-like patterns or provided via a manually labelled training set). However, for tasks wherein one needs to induce hypernymy relationships automatically from a text corpus, Hearst-like patterns usually are not exhaustive.

Yet, there is often a practical requirement in most applications to assign OOV to their most likely correct type(s). Designing a system that fulfills this requirement is highly significant since it allows the creation of hypernymy relationships from a given text corpus, avoiding the problem of sparsity that often characterizes most knowledge bases. The basic idea is to use an augmented SPON approach that leverages distributional similarity metrics between words in the same corpus. This is formally described as follows.

For a given domain, let \(\mathcal{E}_{\text{train}}^{\text{trial}}\) and \(\mathcal{O}_{\text{train}}^{\text{trial}}\) denote the in-vocabulary and OOV input trial hyponym terms; and let \(\mathcal{E}_{\text{test}}^{\text{train}}\) and \(\mathcal{O}_{\text{test}}^{\text{train}}\) denote the in-vocabulary and OOV input test hyponym terms respectively. Let \(\mathcal{E}^{\text{train}}\) denote all the terms observed in the list of training hyponym-hypernym pairs, and let \(\mathcal{H}_{\text{train}}\) denote the list of known hypernyms...
obtained from the list of training pairs. The hyponym terms from $T_{trial}$ and $T_{test}$ are handled by our proposed SPON model, i.e. top-ranked hypernyms for each hyponym term are generated via our model.

The rest of this section deals with how to generate top ranked hypernyms for each hyponym term within $O_{trial}$ and $O_{test}$ respectively. Let $Y_x$ be the random variable denoting the hypernym assignment for an OOV term $x \in O_{test}$ (Similar approach holds for OOV terms from $O_{trial}$). The probability of the random variable $Y_x$ taking on the value $c \in H_{train}$ is then given by,

$$P(Y_x = c|x) = \sum_{h \in H_{train}} P(Y_x = c, h|x)$$

$$= \sum_{h \in H_{train}} P(Y_x = c|h, x).P(h|x)$$

The first equality in the above expression is a direct consequence of Marginalisation property in probability, whereas the second equality merely represents the marginal probability in terms of conditional probability.

We now make a conditional independence assumption, i.e. $Y_x \perp x | h$, or in other words, ignoring the subscript for brevity we have, $P(Y_x = c|h, x) = P(Y = c|h)$. Using this assumption, we can rewrite Equation 12 as,

$$P(Y_x = c|x) = \sum_{h \in H_{train}} P(Y = c|h).P(h|x)$$

$$\approx \sum_{h \in S^p} P(Y = c|h).f(h|x)$$

where $f$ is a scoring function that provides a score between $[0, 1]$, and $S^p$ contains $p$-terms from $E_{train}$ that provide top-k largest values for the scoring function $f$. In practice, we first normalize the values of $f(h|x)$ where $h \in S^p$ using a softmax operation, before computing the weighted sum, as per Equation 13. Also, note that $p$ is a hyper-parameter in this model.

Looking back at Equation 13, we notice that the first part of the summation, i.e. $P(Y = c|h)$, can be obtained directly from our proposed SPON model, since $h \in E_{train}$. In addition, we model the function $f(h|x)$ as cosine-similarity between the vectors for the term $h$ and $x$, wherein the vectors are trained via a standard Word2Vec model (Mikolov et al. 2013) pre-built on the corresponding tokenized corpus for the given benchmark dataset.

Summarizing, given a query OOV term within the trial or test fold of any dataset, our proposed model follows the aforementioned strategy to generate a list of hypernym terms that have been ranked using the formula in Equation 13.

It should be clearly pointed out that our aforementioned proposed OOV strategy is not a stand-alone strategy, rather its performance is inherently dependent of SPON.

6 Unsupervised Benchmarks and Evaluation

SPON is intrinsically supervised because it requires example is-a-pairs for training. However, it can also be applied to unsupervised hypernymy task, provided that example is-a pairs are generated by an external unsupervised process such as Hearst-like patterns.

Table 1: Statistics for benchmark datasets used in unsupervised hypernym detection and direction prediction tasks. The columns represent the number of hyponym-hypernym pairs within the validation and test folds respectively.

| Dataset  | Valid | Test  |
|----------|-------|-------|
| BLESS    | 1,453 | 13,089|
| EVAL     | 736   | 12,714|
| LEDS     | 275   | 2,495 |
| SHWARTZ  | 5,236 | 47,321|
| WBLESS   | 167   | 1,501 |

Table 2: Number of hyponyms in different datasets within SemEval 2018 hypernym discovery task.

|                | English | Italian/ Spanish | Music/Medical |
|----------------|---------|------------------|---------------|
| Train          | 1500    | 1000             | 500           |
| Trial          | 50      | 25               | 15            |
| Test           | 1500    | 1000             | 500           |

Table 1 shows the dataset statistics for unsupervised benchmarks, wherein the split into validation/test folds is already given.

For detection, Average Precision is reported on 5 datasets, namely BLESS (Baroni and Lenci 2011), LEDS (Baroni et al. 2012), EVAL (Santus et al. 2015), WBLESS (Weeds et al. 2014) and SHWARTZ (Shwartz, Goldberg, and Dagan 2016). While for direction, Average Accuracy is reported on 3 datasets, which are BIBLESS (Kiela et al. 2015), BLESS and WBLESS. We refer the readers to Roller, Kiela, and Nickel (2018) for details about these datasets.

We adopted the approach of Roller, Kiela, and Nickel (2018) where a list $L$ of hyponym-hypernym pairs $(x, y)$ is extracted using a Hearst-like pattern-based system. This system consists of 20 Hearst-like patterns applied to a concatenation of Wikipedia and Gigaword corpora, to generate $L$.

Each pair within $L$ is associated with a count $c$ (how often $(x, y)$ has been extracted). Positive Mutual Information (PPMI) (Bullinaria and Levy 2007) for each $(x, y)$ is then calculated. Let the size of $E$ be $m$ and let $M \in \mathbb{R}^{m \times m}$ be the PPMI matrix. We use a similar scoring strategy as Roller, Kiela, and Nickel (2018), i.e. truncated SVD approach to

---

The only exception to this is BIBLESS dataset comprising of 1669 pairs, for which the split is not provided a priori.
generate term embeddings, and score each pair using cosine similarity. This creates a modified list $T$, which is the input for SPON (as mentioned in Section 3).

In order to be directly comparable to Roller, Kiela, and Nickel (2018) and Le et al. (2019), we used the same input file of Roller, Kiela, and Nickel (2018) containing candidate hyponym-hypernym-count triples, i.e. a total of 431,684 triples extracted using Hearst-like patterns from a combination of Wikipedia and Gigaword corpora.

We used the following hyper-parameter configuration for the rank parameter of the SVD based models: 50 for BLEXS, WBLEXS, and BIBLEX; 25 for EVAL, 100 for LEDS and 5 for SHWARTZ. Optimal hyper-parameter configurations for our proposed SPON model were determined empirically using validation fold for the benchmark datasets.

For each experiment, the embedding dimensions $d$ were chosen out of $\{100, 200, 300, 512, 1024\}$, whereas the $\epsilon$ parameter was chosen out of $\{10^{-3}, 10^{-2}, 10^{-3}, 10^{-4}\}$. $k$ is set to 1000 for all experiments. For example, in Table 3, the SPON model used the following hyper-parameters on BLEXS dataset: $d = 300$, $\epsilon = 0.01$.

In addition, we used $L_1$ regularization for model weights, and also used dropout with probability of 0.5. Adam optimizer (Kingma and Ba 2014) was used with default settings. In addition, the term vectors in our model were initialized uniformly at random, and are constrained to have unit $L_2$ norm during the entire training procedure. Furthermore, an early stopping criterion of 20 epochs was used.

### Evaluation

We use the same evaluation script as provided by Roller, Kiela, and Nickel (2018) for evaluating our proposed model. Table 3 shows the results on the unsupervised tasks of hyponym detection and direction prediction, reporting average precision and average accuracy, respectively.

The first row titled Count based (in Table 3) depicts the performance of a Hearst-like Pattern system baseline, that uses a frequency based threshold to classify candidate hyponym-hyponym pairs as positive (i.e. exhibiting hyponymy) or negative (i.e. not exhibiting hyponymy). The ppmi approach in Table 3 builds upon the Count based approach by using Pointwise Mutual Information values for classification. SVD ppmi approach, the main contribution from Roller, Kiela, and Nickel (2018) builds low-rank embeddings of the PPMI matrix, which allows to make predictions for unseen

| Method                        | Average Precision |
|-------------------------------|-------------------|
| OE (Vendrov et al. 2016)      | 0.761             |
| Smoothed Box (Li et al. 2019) | 0.795             |
| SPON (Our Approach)           | **0.811**         |
The Ablation Tests. The analysis in Section 4 which shows that our choice of function $f$ satisfies asymmetry and transitive properties, holds true because $f$ satisfies $f(\vec{x}) \geq \vec{x}$ component-wise. We have chosen to define $f$ as a non-negative activation function $RELU$ followed by a Residual layer. In this section, we perform two sets of ablation experiments, first where we remove the Residual connections altogether, and second where we replace the non-negative activation function $RELU$ with $Tanh$ that can take on negative values.

Table 4 shows the results for each of these ablation experiments, when evaluated on the unsupervised hypernym detection task across four datasets chosen randomly. Removing the Residual layer and using $RELU$ activation function only, violates the aforementioned component-wise inequality $f(\vec{x}) \geq \vec{x}$, and has the worst results out of the three. On the other hand, using Residual connections with $Tanh$ activations may not violate the aforementioned inequality, since, it depends upon the sign of the activation outputs. This argument is supported by the results in Table 4, wherein using $Tanh$ activations instead of $RELU$ almost provides identical results, except for the BLESS dataset. Nevertheless, the results in Table 4 show that encouraging asymmetry and transitive properties for this task, in fact improves the results as opposed to not doing the same.

Furthermore, Table 5 illustrates the results on the unsupervised hypernym detection task for BLESS dataset, wherein we compare our proposed SPON model to other supervised SOTA approaches for hypernym prediction task, namely Order Embeddings (OE) approach as introduced by (Vendrov et al. 2016), and Smoothed Box model as introduced by (Li et al. 2019). We run the OE and Smoothed Box experiments using the codes provided with those papers.

In addition, we used the validation fold within BLESS dataset to empirically determine optimal hyper-parameter configurations, and settled on the following values: For OE, we used an embedding dimensions of 5, generated one negative example for every positive instance using so-called contrastive approach. For Smoothed Box model, we used an embedding dimensions of 50 and generated five negatives per training instance. In either case, we observed that using the entire set of is-a pairs extracted by the Hearst-like patterns (without employing a frequency based cutoff) for training provided the best performance.

From Table 5, it is clear that SPON performs much better (by atleast 1.6%) as compared to Smoothed Box model as well as Order Embedding model in an Unsupervised benchmark dataset.

### 7 Supervised Benchmarks and Evaluation

In the supervised setting, a system has access to a large corpus of text from where training, trial, and test is-a pairs are extracted, and labeled manually.

**Benchmarks**

We used the benchmark of SemEval 2018 Task on Hypernym Discovery. The task is defined as “given an input term, retrieve its hypernyms from a target corpus”. For each input hyponym in the test data, a ranked list of candidate hypernyms is expected. The benchmark consists of five different subtasks covering both general-purpose (multiple languages

| CRIM | MAP  | MRR  | P@5 |
|------|------|------|-----|
| MAP  | 19.78| 36.10| 19.03|
| MRR  | 9.37 | 17.29| 9.19 |
| P@5  | 20.04| 28.27| 20.39|

Table 6: Results on SemEval 2018 General-purpose hypernym discovery task. CRIM, NLP_HZ, and 300-sparsans are the corresponding best systems on English, Spanish and Italian datasets (see Section 2).
### Experimental Settings

This subsection describes the technical solution we implemented for the SemEval tasks, more specifically, the strategies for training dataset augmentation, handling OOV terms, and the hyperparameter optimization.

For the corpora in English, we augmented the training data using automatically extracted pairs by an unsupervised Hearst-like Pattern (HP) based system, following an approach similar to that described by Bernier-Colborne and Barriere (2018), the best system in English, Medical and Music hypernymy subtasks in SemEval 2018. Henceforth, we refer to our pattern-based approach as the HP system.

The HP system uses a fixed set of Hearst-like patterns (e.g., “y such as x”, “y including than x₁, x₂”, etc) to extract pairs from the input corpus. Then, it filters out any of these pairs where either x (hyponym) or y (hypernym) is not seen in the corresponding vocabulary provided by the organizers of the shared task. It also discards any pair where y is not seen in the corresponding gold training data.

Following that, the HP system makes a directed graph by considering each pair as an edge and the corresponding terms inside pair as nodes. The weight of each edge is the count/frequency of how often (x, y) has been extracted by the Hearst-like patterns. It also excludes any cycle inside the graph; e.g. if (x, y), (y, z) and (z, x) then all these edges (i.e. pairs) were discarded. Finally, it discards any edge that has a value lower than a frequency threshold, \(ft\). We set \(ft=10\) for English. For Medical and Music, we set \(ft=2\).

The is-a pairs obtained from HP system are then merged with the corresponding training gold pairs (i.e. treated them equally) to form a larger training set for English, Medical and Music datasets. As a result of this step, the number of total unique training pairs for English, Medical and Music increased to 17,903 (from 11,779 training gold pairs), 4,593 (from 3,256) and 6,282 (from 5,455) correspondingly.

The dataset statistics for the general-purpose and domain-specific hypernym discovery tasks are mentioned in Table 2. It is evident that a significant fraction of the terms in the Trial/Test fold is OOV (see Figure 2), therefore SPON is not able to make any assessment about them. Therefore, in order to handle OOV cases, we represent all terms in the dictionary provided by the SemEval organizers via Word2Vec vectors acquired from the given text corpus.

The dimensions \(d\) for SPON model was chosen from \(\{50, 100, 200\}\), whereas the parameter \(p\) (for handling OOV terms) was chosen from \(\{1, 2, 3, 5, 8, 10\}\). Parameter \(k\) (from Equation 5) was chosen from \(\{100, 200, 500\}\). The regularization, dropout and initialization strategies are exactly similar to Section 7. An early stopping criterion of 50 epochs was used.

### Evaluation

We use the scorer script provided as part of SemEval-2018 Task 9 for evaluating our proposed model. Table 6 shows the results on the three general purpose domains of English, Spanish, and Italian respectively. For brevity, we compare only with the SOTA, i.e., the best system in each task. Performances of all the systems that participated in SemEval 2018 Task on Hypernym Discovery can be found in (Camacho-Collados et al. 2018a). Similarly, Table 7 shows the results on the two domain-specific tasks of music and medical domain corpora. SPON outperforms the SOTA systems in all tasks except for the medical domain in which it achieves comparable results. It is worthwhile to notice that SPON is fully domain-agnostic, i.e., it neither uses any domain-specific approaches, nor any domain-specific external knowledge. We provide an illustration of the output of our system in Table 8, showing a sample of randomly selected terms and their corresponding ranked predictions.

### 8 Conclusion and Future Work

In this paper, we introduced SPON, a novel neural network architecture that models hypernymy as a strict partial order relation. We presented a materialization of SPON, along with an augmented variant that assigns types to OOV terms. An extensive evaluation over several widely-known academic benchmarks clearly demonstrates that SPON largely improves (or attains) SOTA values across different tasks.

In the future, we plan to explore how to extend SPON in two directions. On the one hand, we plan to analyze how to use SPON for the taxonomy construction task (i.e., constructing a hierarchy of hypernyms instead of flat is-a pairs). On the other hand, we plan to generalize our work to relations other than is-a which may have different set of constraints and/or for which pattern-based extraction systems may not exist. For such relations, getting large amounts of training data for a given corpus is a challenge that one needs to circumvent around.
References

Athiwaratkun, B., and Wilson, A. G. 2018. Hierarchical density order embeddings. In ICLR.

Baroni, M., and Lenci, A. 2011. How we blessed distributional semantic evaluation. In Workshop on GEometrical Models of Natural Language Semantics.

Baroni, M.; Bernardi, R.; Do, N.-Q.; and Shan, C.-c. 2012. Entailment above the word level in distributional semantics. In EACL.

Berend, G.; Makrai, M.; and Földiák, P. 2018. 300-sparsans at SemEval-2018 task 9: Hypernymy as interaction of sparse attributes. In SemEval.

Bernier-Colborne, G., and Barriere, C. 2018. CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In SemEval.

Bullinaria, J. A., and Levy, J. P. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods 39(3).

Camacho-Collados, J.; Bovi, C. D.; Anke, L. E.; Oramas, S.; Pasini, T.; Santus, E.; Shwartz, V.; Navigli, R.; and Saggion, H. 2018a. SemEval-2018 Task 9: Hypernymy Discovery. In SemEval.

Camacho-Collados, J.; Delli Bovi, C.; Espinosa Anke, L.; Oramas, S.; Pasini, T.; Santus, E.; Shwartz, V.; Navigli, R.; and Saggion, H. 2018b. SemEval-2018 task 9: Hypernymy discovery. In SemEval.

Dalvi, B. B.; Cohen, W. W.; and Callan, J. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In WSDM.

Fu, R.; Guo, J.; Qin, B.; Che, W.; Wang, H.; and Liu, T. 2014. Learning semantic hierarchies via word embeddings. In ACL.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In COLING.

Kiela, D.; Rimell, L.; Vulić, I.; and Clark, S. 2015. Exploiting image generality for lexical entailment detection. In ACL.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. In ICLR.

Kruszewski, G.; Paperno, D.; and Baroni, M. 2015. Deriving Boolean Structures from Distributional Vectors. Transactions of the Association for Computational Linguistics 3.

Le, M.; Roller, S.; Papaxanthos, L.; Kiela, D.; and Nickel, M. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In ACL.

Li, X.; Vilnis, L.; Zhang, D.; Boratko, M.; and McCallum, A. 2019. Smoothing the geometry of probabilistic box embeddings. In ICLR.

Li, X.; Vilnis, L.; and McCallum, A. 2017. Improved representation learning for predicting commonsense ontologies. In ICML Workshop on Deep Structured Prediction.

Lin, D., and Pantel, P. 2002. Concept discovery from text. In COLING.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In NIPS.

Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. J. 1990. Introduction to WordNet: an on-line lexical database. International Journal of Lexicography 3(4).

Nickel, M., and Kiela, D. 2017. Poincaré embeddings for learning hierarchical representations. In NIPS.

Qiu, W.; Chen, M.; Li, L.; and Si, L. 2018. NLP_HZ at SemEval-2018 task 9: a nearest neighbor approach. In SemEval.

Roller, S., and Erk, K. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In EMNLP.

Roller, S.; Kiela, D.; and Nickel, M. 2018. Hearst patterns revisited: Automatic hypernymy detection from large text corpora. In ACL.

Santus, E.; Yung, F.; Lenci, A.; and Huang, C.-R. 2015. Evaluation 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In Workshop on Linked Data in Linguistics: Resources and Applications.

Shwartz, V.; Goldberg, Y.; and Dagan, I. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In ACL.

Shwartz, V.; Santus, E.; and Schlechtweg, D. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In EACL.

Snow, R.; Jurafsky, D.; and Ng, A. Y. 2004. Learning syntactic patterns for automatic hypernymy detection. In NIPS.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2008. Yago: A large ontology from wikipedia and wordnet. Journal of Web Semantics 6(3).

Tan, L.; Gupta, R.; and van Genabith, J. 2015. Usaar-wlv: Hypernym generation with deep neural nets. In SemEval.

Vendrov, I.; Kiros, R.; Fidler, S.; and Urtasun, R. 2016. Order-embeddings of images and language. In ICLR.

Vilnis, L.; Li, X.; Murty, S.; and McCallum, A. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In ACL.

Weeds, J.; Clarke, D.; Reffin, J.; Weir, D.; and Keller, B. 2014. Learning to distinguish hypernyms and co-hyponyms. In COLING.

Weeds, J.; Weir, D.; and McCarthy, D. 2004. Characterising measures of lexical distributional similarity. In COLING.

Zhitomirsky-Geffet, M., and Dagan, I. 2005. The distributional inclusion hypotheses and lexical entailment. In ACL.